

12-2014

# Interactive Student Planner Application

NII TETTEH TACKIE YARBOI  
drannt@yaho.com

Follow this and additional works at: <http://scholarworks.lib.csusb.edu/etd>

---

## Recommended Citation

TACKIE YARBOI, NII TETTEH, "Interactive Student Planner Application" (2014). *Electronic Theses, Projects, and Dissertations*. Paper 113.

This Project is brought to you for free and open access by the Office of Graduate Studies at CSUSB ScholarWorks. It has been accepted for inclusion in Electronic Theses, Projects, and Dissertations by an authorized administrator of CSUSB ScholarWorks. For more information, please contact [scholarworks@csusb.edu](mailto:scholarworks@csusb.edu).

# INTERACTIVE STUDENT PLANNER APPLICATION

---

A Project  
Presented to the  
Faculty of  
California State University,  
San Bernardino

---

In Partial Fulfillment  
of the Requirements for the Degree  
Master of Science  
in  
Computer Science and Engineering

---

by  
Nii Tetteh Tackie Yarboi

December 2014

# INTERACTIVE STUDENT PLANNER APPLICATION

---

A Project

Presented to the

Faculty of

California State University,

San Bernardino

---

by

Nii Tetteh Tackie Yarboi

December 2014

Approved by:

Dr. Ernesto Gomez, Committee Chair, Computer Science and Engineering

Dr. Josephine Mendoza, Committee Member

Dr. Tong Lai Yu, Committee Member

© 2014 Nii Tetteh Tackie Yarboi

## ABSTRACT

Interactive Student Planner is a web based application that will be used by students of California State University San Bernardino. The proposed system replaces the manual planner given to students by the school. This would allow a student to select courses for a quarter, keep track of homework due dates, create and manage project, create notes, create reports, prompts student for upcoming events, save and keep records of other students. A student can send project report as email to the professor, upload and download files easily. The manual way of storing information is not effective, since the student is limited in what to write and how to keep it safe. The on-line application fixes all these limitations in the manual system. The application is developed in APEX and runs on Oracle 11g as the back-end database. APEX is a data centric and a PL/SQL web application, and the application is built and runs on Windows Server 2012. The system is a mobile web based application and is compatible with most mobile phones irrespective of the operating systems and very responsive on tablets, and pc. In this project the required functionalities of the system will be explained, which includes APEX, web listener , Glassfish Server and how they will help in the deployment of the system. The system will provide a more efficient way of information processing with all the necessary security measures enforced.

## ACKNOWLEDGEMENTS

I would like to thank the almighty God for giving me the strength, knowledge and ability during this important learning stage in my life. My most appreciation goes to my mom, Mrs. Grace Tackie Yarboi, my dad, my siblings and Benjamin Intsiful, an Oracle Apex Certified Expert who supported me to make this educational journey very successful. I would like to express my special thanks to my advisor Dr. Ernesto Gomez, and all the committee members - Dr Tong Yu, and Dr Josephine Mendoza for their reviews, comments and suggestions.

My profound gratitude to Dr. Josephine Mendoza for her advice, nurturing, and her patience. She made me a better student and helped me to achieve my career goals.

This project has developed me into a better professional and have helped me acquired a new technical skills that will help me in my job field.

Lastly, to all the staff and faculty of the School of Computer Science and Engineering, not forgetting Mrs. Monica Latimer and Birdy - may God richly bless you!

## TABLE OF CONTENTS

|  |     |
|--|-----|
| ABSTRACT .....   | iii |
| ACKNOWLEDGEMENTS .....   | iv  |
| LIST OF TABLES .....   | x   |
| LIST OF FIGURES .....  | xi  |
| CHAPTER ONE: BACKGROUND  |     |
| Introduction .....   | 1   |
| Purpose of the Project .....   | 4   |
| Scope of the Project .....   | 5   |
| Significance of the Project .....  | 7   |
| Product Functions .....  | 8   |
| CHAPTER TWO: METHODOLOGY   |     |
| Introduction .....   | 10  |
| Product Perspective .....  | 10  |
| Parties That are Relevant to the System .....                              | 11  |
| Generalizability of Findings .....   | 12  |
| CHAPTER THREE: IMPLEMENTING THE<br>INTERACTIVE STUDENT PLANNER APPLICATION |     |
| Introduction .....   | 14  |
| Architecture / Deployment Diagram .....                                    | 15  |
| Application Implementation .....   | 16  |
| The Student .....  | 16  |

|  |    |
|--|----|
| Tabs .....   | 16 |
| System .....   | 17 |
| Scalability .....  | 17 |
| Availability .....   | 17 |
| Application Flowchart .....                                | 18 |
| Homepage .....   | 18 |
| Courses .....  | 20 |
| Homework .....   | 22 |
| Project .....  | 24 |
| Contacts .....   | 27 |
| Reports .....  | 30 |
| Notes .....  | 32 |
| Events .....   | 34 |
| Resources .....  | 36 |
| High Level System Architecture Of Apex .....               | 38 |
| Design and Implementation Constraints .....                | 42 |
| The Application Programming .....                          | 42 |
| The Modules of the System Architecture .....               | 42 |
| Front-end Development .....                                | 43 |
| Stand Alone (What Needs To Be Installed By The User) ..... | 43 |
| Back-end Development .....                                 | 44 |
| Web Service (Maintenance And Administration) .....         | 44 |



|                                 |    |
|---------------------------------|----|
| Database Design .....           | 46 |
| User Interface .....            | 52 |
| Homepage .....                  | 54 |
| Courses .....                   | 55 |
| Homework .....                  | 56 |
| Project .....                   | 57 |
| Contacts .....                  | 60 |
| Reports .....                   | 62 |
| Notes .....                     | 64 |
| Events .....                    | 66 |
| Resources .....                 | 68 |
| <br>CHAPTER FOUR: DATA SECURITY |    |
| Introduction .....              | 70 |
| Types of Data Security .....    | 72 |
| Application Security .....      | 72 |
| URL Tampering .....             | 72 |
| Authentication .....            | 73 |
| Authorization .....             | 73 |
| Page Flow Control .....         | 74 |
| Validations Processing .....    | 75 |
| User .....                      | 75 |

|  |     |
|--|-----|
| Server(Glassfish) .....  | 77  |
| Database Security .....  | 78  |
| Auditing .....   | 78  |
| Row Level Security .....   | 79  |
| SQL Injection .....  | 80  |
| Cross-Site Scripting (XSS) .....   | 82  |
| Encryption .....   | 82  |
| Secure Views .....   | 84  |
| Table-Level Security .....   | 84  |
| Some Other Components to Setup to Achieve Security .....                   | 85  |
| Router .....   | 85  |
| Firewall .....   | 85  |
| Apex Listener .....  | 85  |
| CHAPTER FIVE: CONCLUSION   |     |
| Introduction .....   | 86  |
| Findings .....   | 87  |
| Lessons .....  | 90  |
| Benefits .....   | 91  |
| Future Work .....  | 92  |
| APPENDIX A: INSTALLING ORACLE DATABASE<br>11G ON WINDOWS SERVER 2012 ..... | 93  |
| APPENDIX B: INSTALLING APEX 4.2 .....                                      | 96  |
| APPENDIX C: INSTALLING GLASSFISH SERVER .....                              | 100 |

|   |     |
|---|-----|
| APPENDIX D: DATABASE OBJECT SUMMARY<br>AND HOW TO CREATE THE SCHEMA ..... | 113 |
| APPENDIX E: HOW TO CREATE THE TRIGGERS .....                              | 116 |
| APPENDIX F: HOW TO CREATE THE FUNCTIONS .....                             | 122 |
| APPENDIX G: CODE FOR SOME SECTION OF THE HOMEPAGE .....                   | 125 |
| APPENDIX H: CODE FOR SOME SECTION OF THE COURSE PAGE .....                | 129 |
| APPENDIX I: CODE FOR SOME SECTION<br>OF THE PROJECT SUMMARY PAGE.....     | 131 |
| APPENDIX J: CODE FOR SOME SECTION<br>OF THE PROJECT DETAILS .....         | 135 |
| APPENDIX K: CODE FOR SOME SECTION<br>OF THE CONTACT PAGE .....            | 139 |
| APPENDIX L: CODE FOR SOME SECTION<br>OF THE NOTE SUMMARY PAGE .....       | 141 |
| APPENDIX M: CODE FOR SOME SECTION<br>OF EVENT CALENDAR PAGE .....         | 145 |
| REFERENCES .....  | 147 |

## LIST OF TABLES

|  |    |
|--|----|
| Table 1. Sample Apps .....   | 3  |
| Table 2. Comparisons of the Two Systems .....                                    | 11 |
| Table 3. Tentative Schedule of Tasks Need to Carry out<br>By The Developer ..... | 13 |
| Table 4. Data Dictionary of the Database Tables .....                            | 48 |

## LIST OF FIGURES

|   |    |
|---|----|
| Figure 1. Use Case Diagram .....  | 9  |
| Figure 2. Standard Apex Architecture .....                                    | 15 |
| Figure 3. Flowchart of Homepage .....   | 19 |
| Figure 4. Flowchart of Courses Page .....                                     | 21 |
| Figure 5. Flowchart of Homework Page .....                                    | 23 |
| Figure 6. Flowchart of Project Page .....                                     | 26 |
| Figure 7. Flowchart of Contact Page .....                                     | 29 |
| Figure 8. Flowchart of Report Page .....                                      | 31 |
| Figure 9. Flowchart of Note Page .....  | 33 |
| Figure 10. Flowchart of Event Page .....                                      | 35 |
| Figure 11. High Level System Architecture .....                               | 38 |
| Figure 12. F-Procedure of Apex (Heart of Apex) .....                          | 41 |
| Figure 13. Entity Relationship Diagram .....                                  | 49 |
| Figure 14. Unified Modeling Language Diagram .....                            | 51 |
| Figure 15. User Login Screen .....  | 53 |
| Figure 15a. Change of Password Page .....                                     | 53 |
| Figure 16. Desktop/Mobile Interface of Homepage .....                         | 54 |
| Figure 17. Desktop/Mobile Interface of<br>Course Page with Lookup Table ..... | 55 |
| Figure 17a. Desktop/Mobile Interface of Course Page .....                     | 55 |
| Figure 18. Desktop/Mobile Interface of Homework Page .....                    | 56 |
| Figure 18a. Desktop/Mobile Interface of Homework Report .....                 | 56 |

|   |    |
|---|----|
| Figure 19. Desktop/Mobile Interface of Project Page .....                   | 57 |
| Figure 19a. Desktop/Mobile Interface of Project Details Page .....          | 58 |
| Figure 19b. Desktop/Mobile Interface of Send Email in Project Page .....    | 58 |
| Figure 19c. Desktop/Mobile Interface of Create Project Page .....           | 59 |
| Figure 19d. Desktop/Mobile Interface of Edit Project Page .....             | 59 |
| Figure 20. Desktop/Mobile Interface of Contact Page .....                   | 60 |
| Figure 20a. Desktop/Mobile Interface of Upload Contact .....                | 60 |
| Figure 20b. Desktop/Mobile Interface of Add New Contact .....               | 61 |
| Figure 20c. Desktop/Mobile Interface of Edit Contact .....                  | 61 |
| Figure 21. Desktop/Mobile Interface of Report Page .....                    | 62 |
| Figure 21a. Desktop/Mobile Interface of Summary Page .....                  | 62 |
| Figure 21b. Desktop/Mobile Interface of Homework Report .....               | 63 |
| Figure 21c. Desktop/Mobile Interface of Homework Scores .....               | 63 |
| Figure 22. Desktop/Mobile Interface of Note page .....                      | 64 |
| Figure 22a. Desktop/Mobile Interface of Create Note .....                   | 65 |
| Figure 22b. Desktop/Mobile Interface of Edit Note .....                     | 65 |
| Figure 23. Desktop/Mobile Interface of Event Page .....                     | 66 |
| Figure 23a. Desktop/Mobile Interface of Create Event .....                  | 67 |
| Figure 23b. Desktop/Mobile Interface of Edit Event .....                    | 67 |
| Figure 24. Desktop/Mobile Interface of Resources Page .....                 | 68 |
| Figure 24a. Desktop/Mobile Interface of Departments in Resources Page ..... | 68 |
| Figure 24b. Desktop/Mobile Interface of Important Dates .....               | 69 |

|  |    |
|--|----|
| Figure 24c. Desktop/Mobile Interface of Campus Map .....                                     | 69 |
| Figure 25. Scope of Improving Web Application Security:<br>Threats and Countermeasures ..... | 71 |
| Figure 26. User Access Control Login Page .....  | 76 |
| Figure 27. Pie Chart of Students Sample .....  | 87 |

## CHAPTER ONE

### BACKGROUND

#### Introduction

California State University San Bernardino (CSUSB) gives out an annual student planner through the Associated Students Incorporated. The students organize their activities for each quarter. This is certainly a good idea, but in this technological age keeping records in a flat file is so outdated, costly, not secured, and restrictive. It is restrictive because students are limited to the number of pages for notes, and the features for a particular section is lessened. A lot more functions can be added to this planner to make it very interactive, If the school is supposed to print about 10000 copies every year, it will cost them a lot of money. That money can be used to create different resources that will benefit students. Each student has to keep this flat file every year, the probability of it getting lost or stolen is very high. Imagine the student losing the student planner which contains some personal information?







After conducting an interview with students, a majority said they wanted a means by which they can check the due dates of homework with a simple click of a button or easy access to a mobile device. The students sometimes forget the due dates and mostly remember on the last day of submission. Other students would also wish there was an electronic way of keeping contacts, like a year book, where the student can keep photos, names, addresses, and phone



numbers of friends. The flat file way of storing this information was daunting due to some of the constraints mentioned above. The students will also want to keep and print records of classes, as well as be updated on upcoming events offered by CSUSB. Finally, others said they would like to keep track of homework scores and know how they are doing in a course. After taking CSE680 Distributed Database Management Systems and my knowledge on Oracle along with my passion for databases, I found out that creating an application in Oracle Apex will help fix a lot of the issues dealing with a manual planner.

Several apps have been created such as Inktopad Notepad, Colornote, Classic Notes, Note Everything, Notes, Cute Notes, Quick Notes, Calendar, Timetable, Student Agenda, MyHomework Student Planner, Student Timetable Helper, Class Buddy, and a whole lot. C-net list the best 7 great Android apps for keeping track of notes, tasks, and to-do lists on <http://www.cnet.com/news/7-great-android-apps-for-notes-and-tasks/>

Table 1. Sample Apps

| APP   | NAME                             | O.S             | FEATURES  | REFERENCES: WEBSITES  |
|---|----------------------------------|-----------------|---|---|
|    | Notes                            | iOS             | Notes can be synced via iTunes and show up in your Mail application. You can also email them right out of the mobile app.   | <a href="http://iphone.appstorm.net/roundups/productivity-roundups/13-notable-note-taking-apps-for-iphone/">http://iphone.appstorm.net/roundups/productivity-roundups/13-notable-note-taking-apps-for-iphone/</a>                             |
|    | Note Taker                       | iOS             | This app takes a very different approach in actually taking down a note: you can write on the screen of your iDevice with your finger and the app will take your scribbles down as a note | <a href="http://iphone.appstorm.net/roundups/productivity-roundups/13-notable-note-taking-apps-for-iphone/">http://iphone.appstorm.net/roundups/productivity-roundups/13-notable-note-taking-apps-for-iphone/</a>                             |
|    | Homework Planner                 | Android and iOS | Grade tracking, timetabling and various other features  | <a href="https://play.google.com/store/apps/details?id=com.tom.hwk">https://play.google.com/store/apps/details?id=com.tom.hwk</a>   |
|    | Power Planner (Homework Planner) | windows         | With live tiles, automatic reminders, multiselect lists, a grading system, and online sync,   | <a href="http://www.windowsphone.com/en-us/store/app/power-planner-homework-planner/46a9c430-f253-4043-99c6-bc600335e9de">http://www.windowsphone.com/en-us/store/app/power-planner-homework-planner/46a9c430-f253-4043-99c6-bc600335e9de</a> |
|   | Homework Planner                 | Android         | Allows user to record the subjects that you have homework in by using a check box system. In the assignment box you can log homework assignments via speech to text or manual text input. | <a href="https://play.google.com/store/apps/details?id=appinventor.ai_tylerleonard94.homeworkplanner">https://play.google.com/store/apps/details?id=appinventor.ai_tylerleonard94.homeworkplanner</a>   |
|  | Touch Calendar                   | iOS             | A user see your whole calendar laid out in front of you and you can double-tap or pinch-to-zoom to focus on a specific section, week, or day. It's quite a cool concept.                  | <a href="http://iphone.appstorm.net/roundups/productivity-roundups/13-notable-note-taking-apps-for-iphone">http://iphone.appstorm.net/roundups/productivity-roundups/13-notable-note-taking-apps-for-iphone</a>                               |

The interactive student planner app is different in that it will integrate most of the mentioned apps listed in [e.g., (see Table 3)], into one single app, but will provide more functionalities. The app will have the flexibility of running on a pc, as well as on any smart phone or tablet. The system framework is designed to run on any mobile device irrespective of the operating system. The student does not need to worry about updates such as font size, or color changes, added new

features, because the updates are uploaded to the application server and becomes immediately available for all users of the system. The application will also have a link for contacts, where the student can keep records of past and current students with their major, phone number, and email for easy communication. The app will help keep track of alumni and make it easy for students to stay in touch. The more procedural PL/SQL which runs on an built in database will be used to create and manage the app.

### Purpose of the Project

The purpose of this project is to create an Oracle APEX application for CSUSB, known as interactive student planner that would:

- Add a planner (daily, weekly, monthly, or yearly)
- Keep records of the courses to be taken or have been taken over a quarter
- Keep track of homework and project
- Write and save notes
- Keep other students information, e.g. contacts
- Save and email a project report to professors
- Keep records of all student electronic files for quick retrieval

- Prompts for upcoming events, such as workshops
- Keep reports and resources to help the student with relevant information

### Scope of the Project

The application allows a student to create a planner based on specific fields, and the information generated will be saved as the planner. The student will have the flexibility to edit a field if there is a mistake. A student can make and save notes on each planner, can easily search for particular information based on the field and the fields can be filtered for easy searches.

Each student will be able to keep records of the courses to be taken or have been taken over a quarter. The student has the ability to edit or add any field of preference. This will make it easier to track all courses needed for graduation.

Since almost every class at CSUSB assigns a project, the project link will help monitor project status over time with more functions to make organizing better for students. A project in the planner will have a number of attributes, including one or more owners (for group projects), description, and goal. A student can submit status updates for a project and create status report based on a selection of updates. Projects are assigned to a category, and a project can be an individual or a group project. Although all projects will allow a student to add files, the size will be limited so as to restrict the student from adding much larger files, more than 15M. The attached files contain information relating to a project.

Each project will have a description, and can be updated to indicate progress of a particular project . A student can easily send reports to an instructor or other group project members through emails, and track progress for a given project.

The student will have the ability to keep records of homework, due dates, scores and will be able to upload homework in any format such as pdf, word, jpg etc. This section will include a course name, course number, and the homework description. There is a donut chart to graphically present the average sum of homework scores. This is an important feature to help the student to know the progress in a particular course. Each homework will show up on the home screen with the due dates counter as shown in [e.g., (see Figure 16)].

A student will be able to create and save the contact details of other students, their majors and professions. This will be an advanced form of a year book, that will help most students to track others and get job opportunities based on their majors. The application will also prompt a student of upcoming events offered at CSUSB.

The University will still have some pages assigned to the designated members, who write some useful information in the planner, this will include but not limited to post welcome messages , and other relevant information. There is a feature known a resources which will contain student life at CSUSB, campus Orgs, departments, important dates, campus map, other campus, restaurants, shopping, attractions, campus planning, guide, first aid, writing strategies. These

will ensure that some useful and significant information in the manual system of planner will not be discarded. Moreover the map on a mobile phone or tablet will make it very convenient and easy to use by most students, rather than holding a planner. The administrator will only have access to upload the messages and suggestions from the school because the administrator has access to the backend of the application.

### Significance of the Project

The project will help the student to handle personal information in an effective way. The most beneficiary of the project is the student, the project will provide the necessary information for a class on a mobile device, hence the student can keep track of classes and focus easily on their academic goals. The average student spends most of their time online, involved in some social activities, for this reason the need for an interactive student planner will be accepted by most students. This project would generate a great deal of interest to the student of California State University San Bernardino and also any other student interested to use the application.

## Product Functions

With the web application the student will be able to perform more functions than the manual planner. There are several tabs for the student to navigate to a page of interest and perform whatever task they want to, based on the description and setup on the page component. Each student will have a unique user name and password to access their planner. These passwords will be given to the students by the administrator, when the student logs into the system, there is an option to change the password to their requirements. It's only an administrator who will have the privileges to alter a table, drop a table or add constraints.

## Use Case

The student will have constant interaction with the system

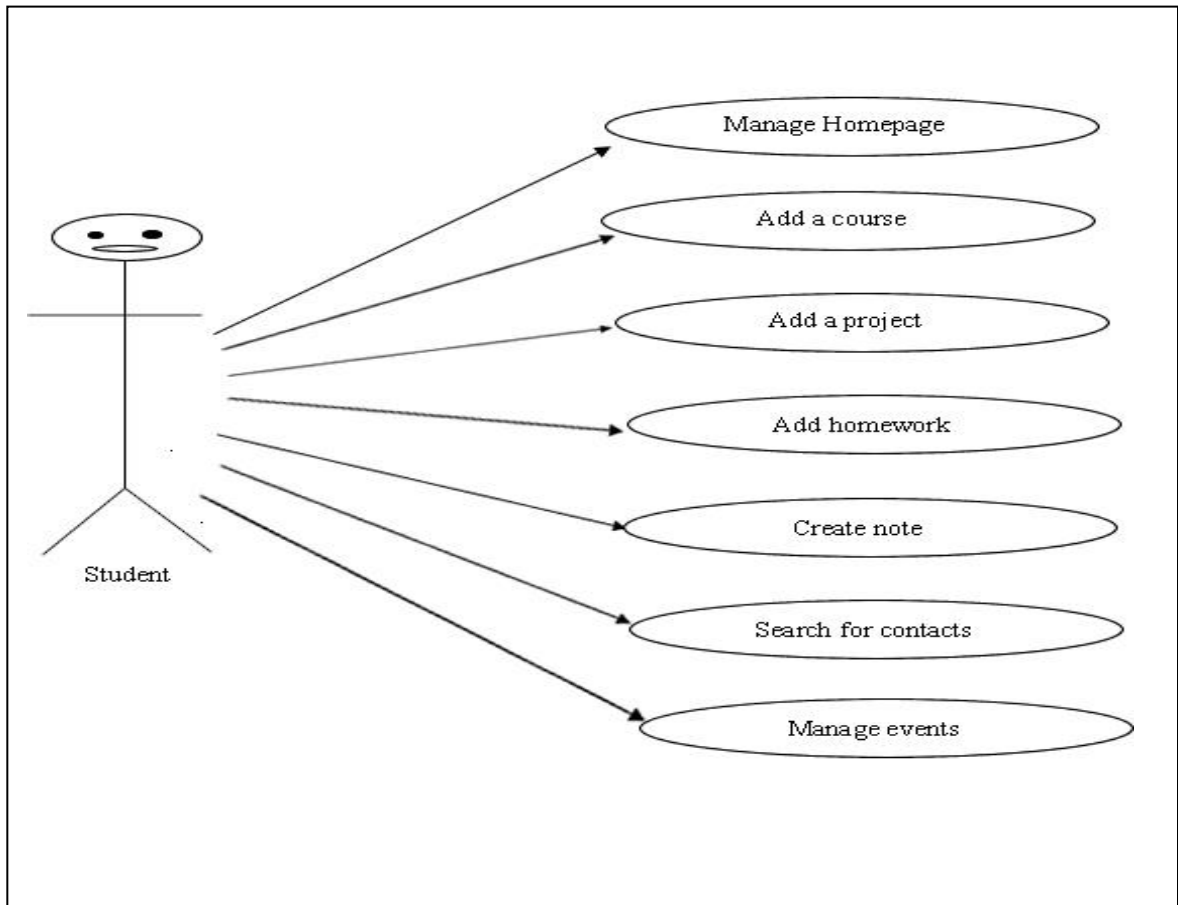


Figure 1. Use Case Diagram



## CHAPTER TWO

### METHODOLOGY

#### Introduction

Although APEX has built in themes, templates, and forms, this system consist of personalized theme, personalized buttons, a database, and views for the users. It requires PL/SQL, JavaScript, CSS, HTML to generate and manipulate the pages. This will give the developer full control of what the page is required and the business rule it has to meet, in order to satisfy the user requirements . Much effort will be placed on the database end to meet the Atomicity, Consistency, Isolation and Durability of a database.

Security, Performance, Scalability, Upgrades, Transaction Control, Packaging, Re-use and Web services will all be done by the developer with some extensive database administrative knowledge.

#### Product Perspective

The Student Planner Application will replace the existing one which is the manual planner, given to the students at the beginning of each quarter. Almost every student has the same planner and the student need to write out his/her plans in it. With this new product everything will be computerized and information retrieval will be faster and much secured.

Table 2. Comparisons of the Two Systems

| MANUAL SYSTEM                                | INTERACTIVE STUDENT PLANNER  |
|--|--|
| Limited number of pages                      | Electronic, so the student can write as much, each student is assigned a specified space |
| Not secured                                  | Password protected   |
| Easily lost                                  | Information is on the server, the student just logs on.                                  |
| Not mobile enough                            | Application can be accessed on a mobile phone or any computer                            |
| Costly to print more copies for each student | All the student needs is availability to a mobile phone or a web browser                 |
| No better Archiving                          | Oracle database means data can be saved for a long period of time                        |

#### Parties That are Relevant to the System

The main parties that are relevant to the system are the student and the administrator. The Administrator will be in charge of any code modification, database backup and all the administrative task, whereas the student, the main party will only sign up to use all the functionalities of the system to meet their specific goals.

The tabs of the student planner app

- The Homepage
- Courses
- Homework
- Project
- Contacts
- Notes
- Events
- Reports
- Resources

### Generalizability of the Findings

The application is targeted towards computer science students of California State University San Bernardino, in view of that the courses in the lookup course table are courses offered by the departments. Other students can use the course section but the students have to enter the courses manually. The student can filter to select specific courses based on the quarter and the class number.

Table 3. Tentative Schedule of Tasks Need to Carry Out  
By The Developer

| TASKS  |
|--|
| Installing Oracle Database 11g   |
| Installing Apex  |
| Create Tablespace  |
| Setup Listener   |
| Set password for Administrator   |
| Create workspace   |
| Create schema  |
| Create Tables  |
| Create Triggers, Sequences, Normalized Tables                            |
| Create Application   |
| Create my own responsive Theme   |
| Create each page( coding in PL/SQL, CSS. Creating and developing pages ) |
| Connecting to DB   |
| Setting up Glassfish   |
| Configuring the application server for the administrative privileges     |
| Deploying the weblistener on the glassfish server                        |
| Getting Apex to connect through Glassfish                                |

## CHAPTER THREE

### IMPLEMENTING THE INTERACTIVE STUDENT PLANNER APPLICATION

#### Introduction

The Interactive Student Planner is a database application which runs on an Oracle database as the backend. Although Oracle allows free download of the enterprise edition, it is not allowed to use for production purposes. It is expensive to purchase the Oracle Enterprise database software, but the university already runs an Oracle database, which will support in the development and hosting of the application. Apex can be used in some production fields as Software as a Service (SaaS) applications on some cloud providing services, such as Salesforce.com. The applications that a user develops are hosted and run from Salesforce.com servers, much of the work is done for the user and there is no need to worry about hosting, but these come with a cost. Anyone interested in developing an application can sign in to Salesforce and use their client server and database servers. The proposed application will not be deployed on the cloud but hosted on the university database server. The system is built on the standard architecture of APEX from Oracle but a more higher level architecture will be introduced to boost security.

## Architecture / Deployment Diagram

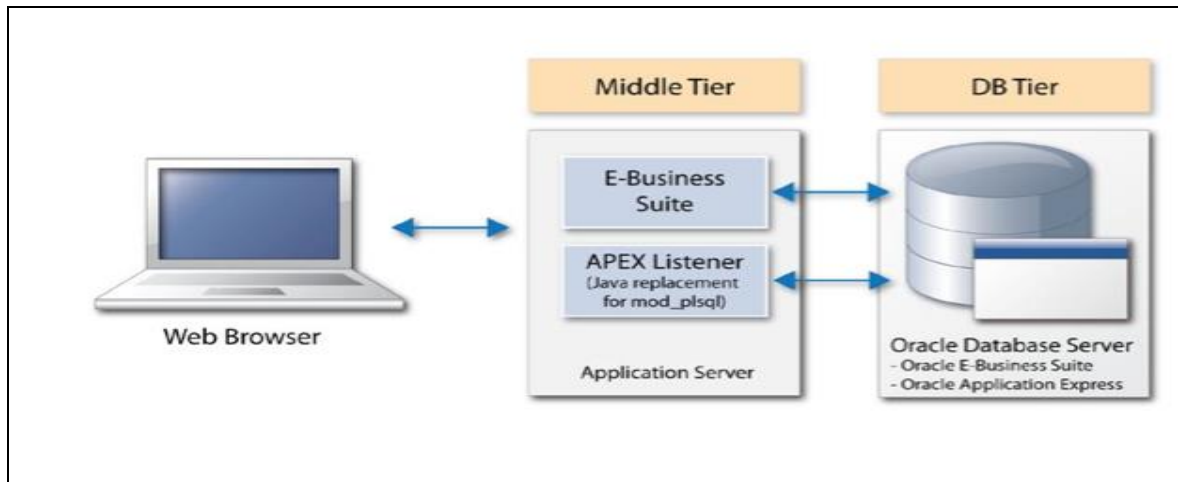


Figure 2. Standard Apex Architecture

Retrieved from <http://www.oracle.com/technetwork/developer-tools/apex/apex-arch-086399.html>

Apex is a development tool based on an Oracle database, the system will be deployed on the 3 tier architecture. An end user (student) will use his web browser to go to any apex application by entering a URL into the browser, the request is sent to the listener, the listener recognizes it as apex request and sends it to the apex engine in the oracle database. The apex engine uses metadata and its own schemas combined with other data in the custom schemas to generate the contents of html web page which is then forwarded back out to the web listener and to the client.

## Application Implementation

The main entity of the system is the student who will be the primary user of the application. Some other systems that are not of much concern for the student are the Oracle database and the web server which will be managed by the administrator. Various parameters and installation procedure must be met in order to achieve a complete system functionality.

### The Student

The User of the system can connect to the system either via an Administrator created password / Sign up, both give access to the change of password, a notification is received that account has been created and must be activated. The former is implemented in the system, in order to closely monitor if registered students are those using the system, the student can create biographic information, even though not necessary. The sign up is an option but will be used in the future when the application will allow any other student outside CSUSB to use the system.

### Tabs

There are 8 tabs which are link to several pages in the application, these give the student the flexibility to navigate through his/her planner. Without flipping pages, or handling the daunting task of a limited number of pages the student has unlimited amount of space to write and save notes.

## System

The main feature of the system is to make sure that required fields are entered accordingly. The system will prompt the student of errors when input is not valid or some data is missing in the required fields. The buttons perform the specific actions depicted in the labeling, but the system will prompt the student of some actions which are not reversible like delete, apply changes. The database and the administrative work will be done at the back end.

## Scalability

The student planner applications can scale indefinitely to support additional users, without having to deploy additional servers. The application will run faster because a single query can obtain information from multiple objects.

## Availability

The application will be up and running all the time, unless in cases where the server is shut down by the administrator, or there are problems pertaining to the end user.



## Application Flowchart

### Homepage

The homepage is a summary of the application in a dashboard, it consists of the major features of the student planner

- The total number of projects is displayed.
- Total Courses shows a summary of the courses the student is enrolled in a quarter. It is linked to the course page.
- Total Homework shows the number of homework in a quarter

- Homework due dates shows the homework, description and due dates.

The due date color is green when the dates is not due, turns yellow when it's getting due and red when it is finally due. This will prompt the student that a homework needs attention and the student can attend to it.

- The student can edit the homework, save it or delete it. The homework serves as a reminder and can be easily accessed on a smart phone.
- The calendar is a representation of the students' upcoming events, location and time. It is linked to the database which supply the required data entered by the student.
- The donut chart shows the average total scores of the homework based on a different course.

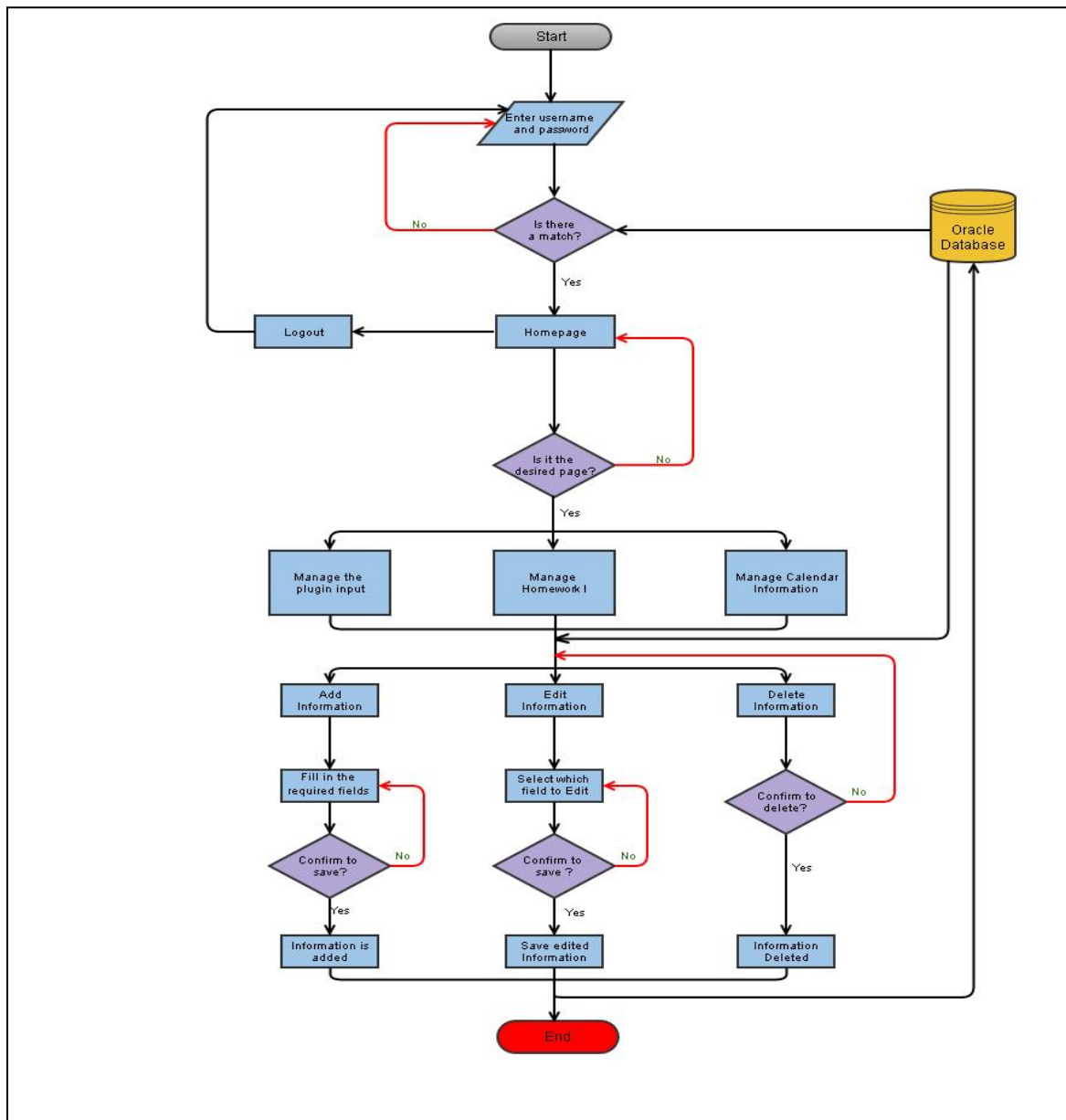


Figure 3. Flowchart of Homepage

## Courses

- The student can add the course manually or can access it from the courses lookup table.
- Courses lookup table will contain the courses offered at CSUSB Computer Science Department. A student can access the course by going through the lookup table, the student adds the specific course to a course cart, which holds the courses.
- The student can filter the search to narrow the specific search criteria, either by the course name, course number, quarter.
- The final step is to confirm the course in the student course table, this shows the selected course for a quarter. A student can edit the course, this course is what is displayed on the homepage.

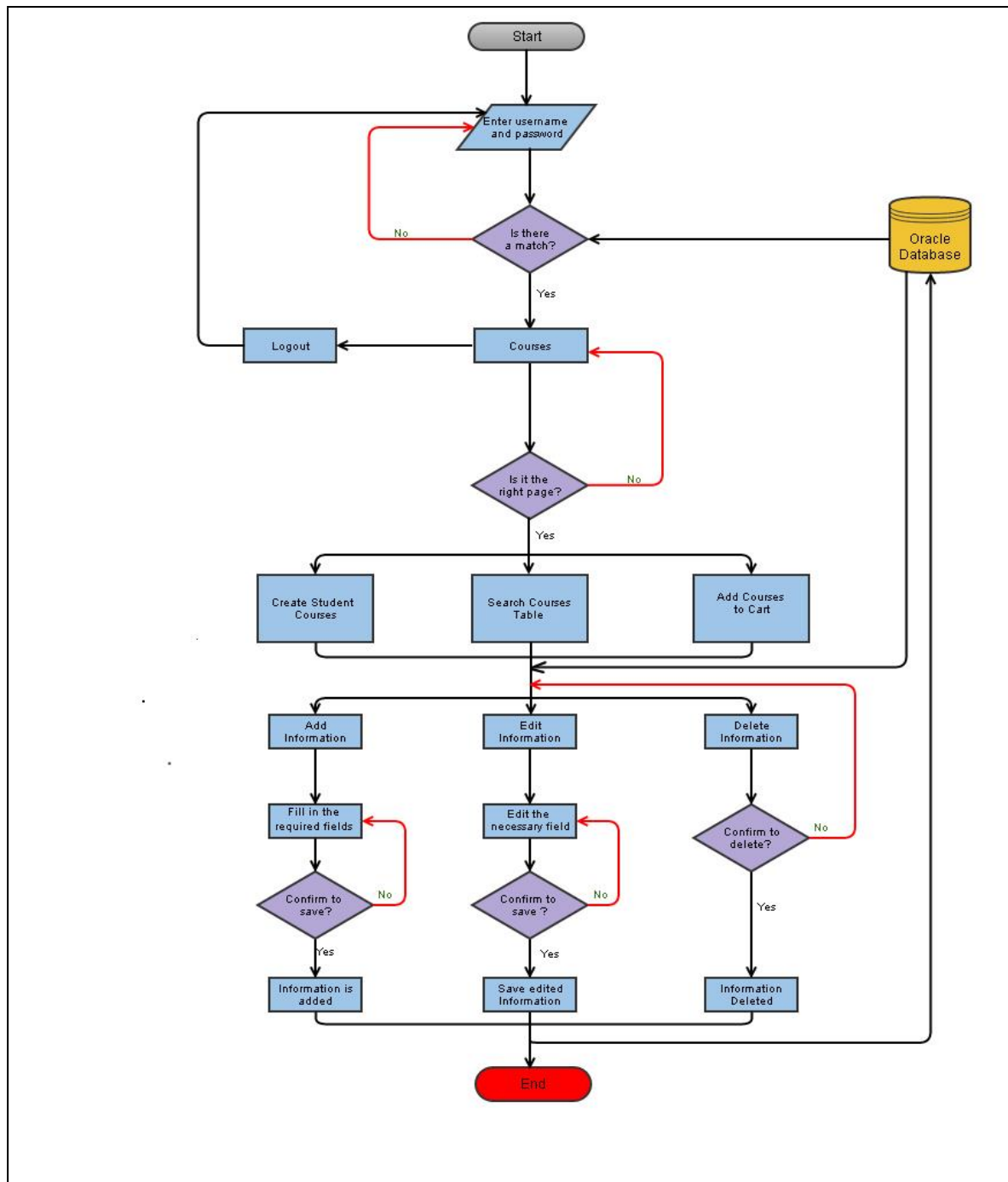


Figure 4. Flowchart of Courses Page

## Homework

- The student keeps track of due dates of homework, and can edit the fields easily.
- The student enters the homework name, description, due date and some actions to be carried out.
- The homework description is a detailed information or a name of the homework to differentiate each homework. It is linked to the homework table
- The course name shows which homework is attached to a particular course.
- When the student creates the homework, it is added to the saved homework table report. the student can perform some actions on the table, including sorting, downloading, saving, flashback..
- The student also has the ability to and enter a particular homework to edit the homework name, description, due date, and course name. the student can click on the apply changes button or can delete an entry.

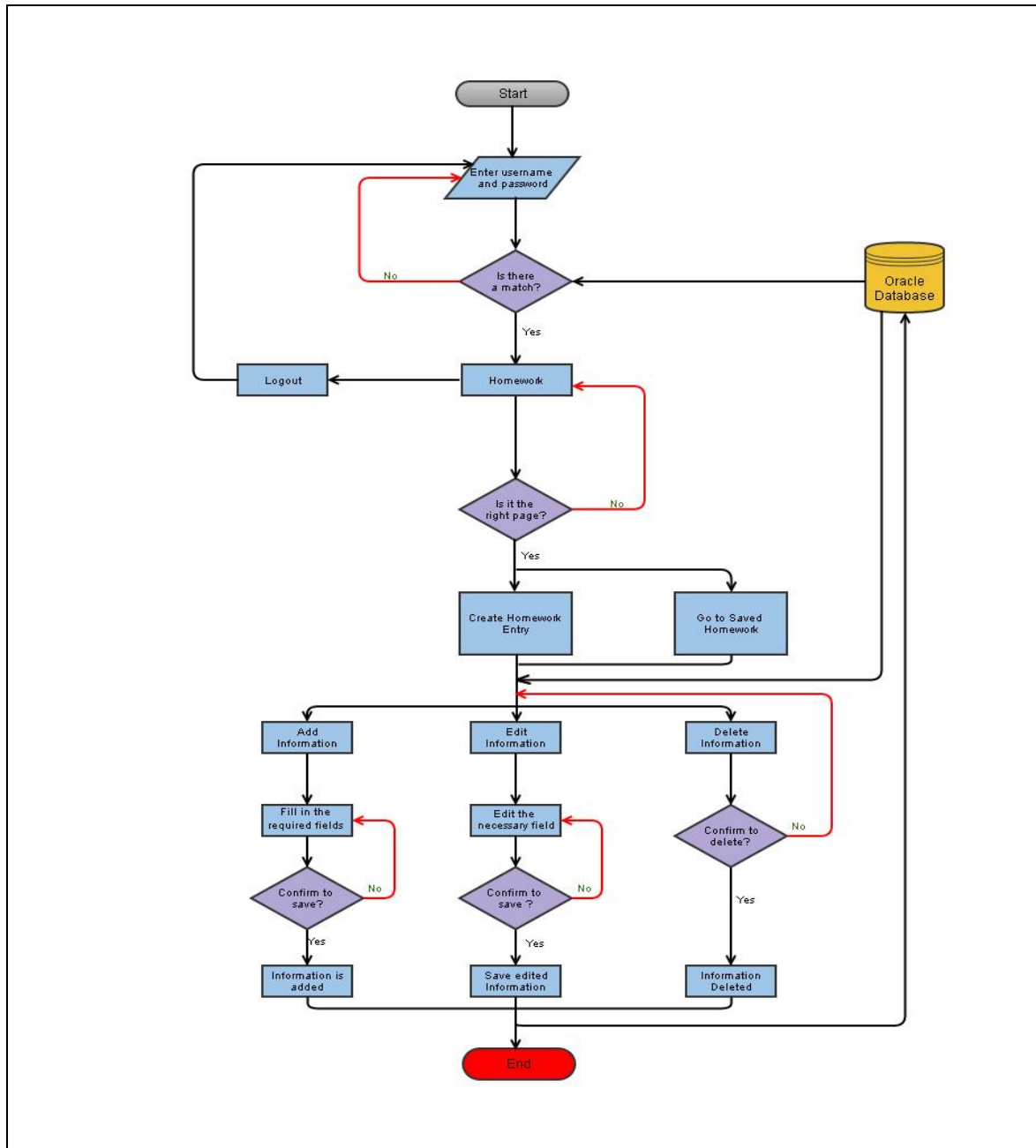


Figure 5. Flowchart of Homework Page

## Projects

- The student will track the progress of his project, print, upload and download files associated with a particular project
- The project page consists of a list of projects, last updated, due date, description. The list of projects is in order, the last updated shows the last time a particular project was work on. The description is the full detail of what the project is about.
- The print button allows the student to print the project list and details.
- To create a new project, the student clicks on the create project button, this takes the user to a screen with customized fields, such as project name, start date which is required, the student cannot proceed without filling out those options.
- The project description is the detailed explanation of the project, it is exactly what the user project will entail.
- When the user clicks on cancel, the application is returned to the project homepage.
- When the student clicks on add a project, the details are loaded into the project table. The project table consist of project names.

- The user clicks on a particular project, it goes to the project details page, where the student can edit the page details. or add an attachment to a project.
- To add the attachment, click on the addfile, this takes the student to the file upload page. The student can select a specific project from the drop down based on the current project the student have at a moment.
- The browse buttons gives the student the option to select a file, the files could be any format, but must be under 15M in size, a trigger is created to detect this parameter before any insertion into the database.
- The student can include file comments to help identify the file, and can add files to the project attach tables or cancel.
- The project attachment table shows the filename, type, when it was created , who created it, the file size.
- The student has the ability to download a file to his phone or laptop, this can be sent as an attachment or can be saved.
- The student can reset the report, but will lose every detail and each project can be emailed to a group member or the professor.
- The student has the ability to edit a project attachment, the student can cancel, edit or apply changes.



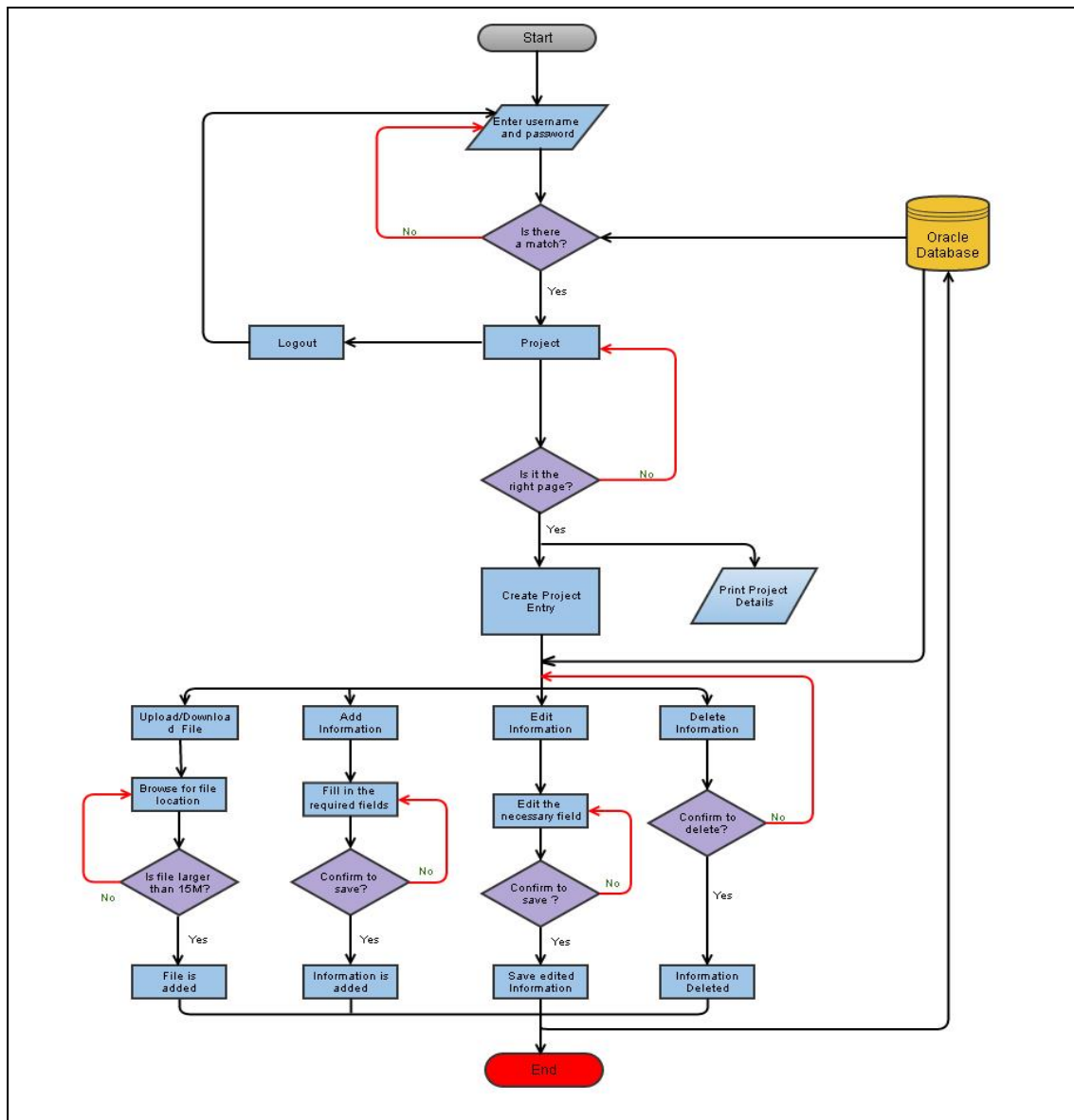


Figure 6. Flowchart of Project Page

## Contacts

- This is like the yearbook, where the student can create the specific fields: first\_name, last\_name, gender, email and photo attachment for easy identification.
- The student can easily update his contacts by uploading the file in an excel form, hence no need to manually enter all contact details
- the student can go into a particular field and edit, make changes in the required fields.
- The load data button allows a student to upload an excel or csv file, this makes it easy for the application to be used with the contacts on the phone, once the contacts is saved in the vcf on phones it can be converted into excel or csv which will then be uploaded easily into the application. This will also help in the manual entry on data. Once the student import contacts and their information they can be uploaded easily without having to create the members individually, this saves the time.
- The upload contact page is made up of 4 pages, the first page gives a default button in uploading the file, the student then browse to select a file. The character '\t' is used separate a column, but a comma will be used to perform the same function.

- Optionally enclosed helps to ignore white spaces and marks the starting and the ending boundary of a data value.
- First Row has Column Names: this box is selected if the student data contains column name in the first row.
- Data/ Table mapping is where the student matches the column for the database and his extended uploaded files.
- All columns must be matched in order for the next insertion to be effective, otherwise not all the columns will be inserted.
- Data valid is way the system confirms of the data manipulation that has occurred

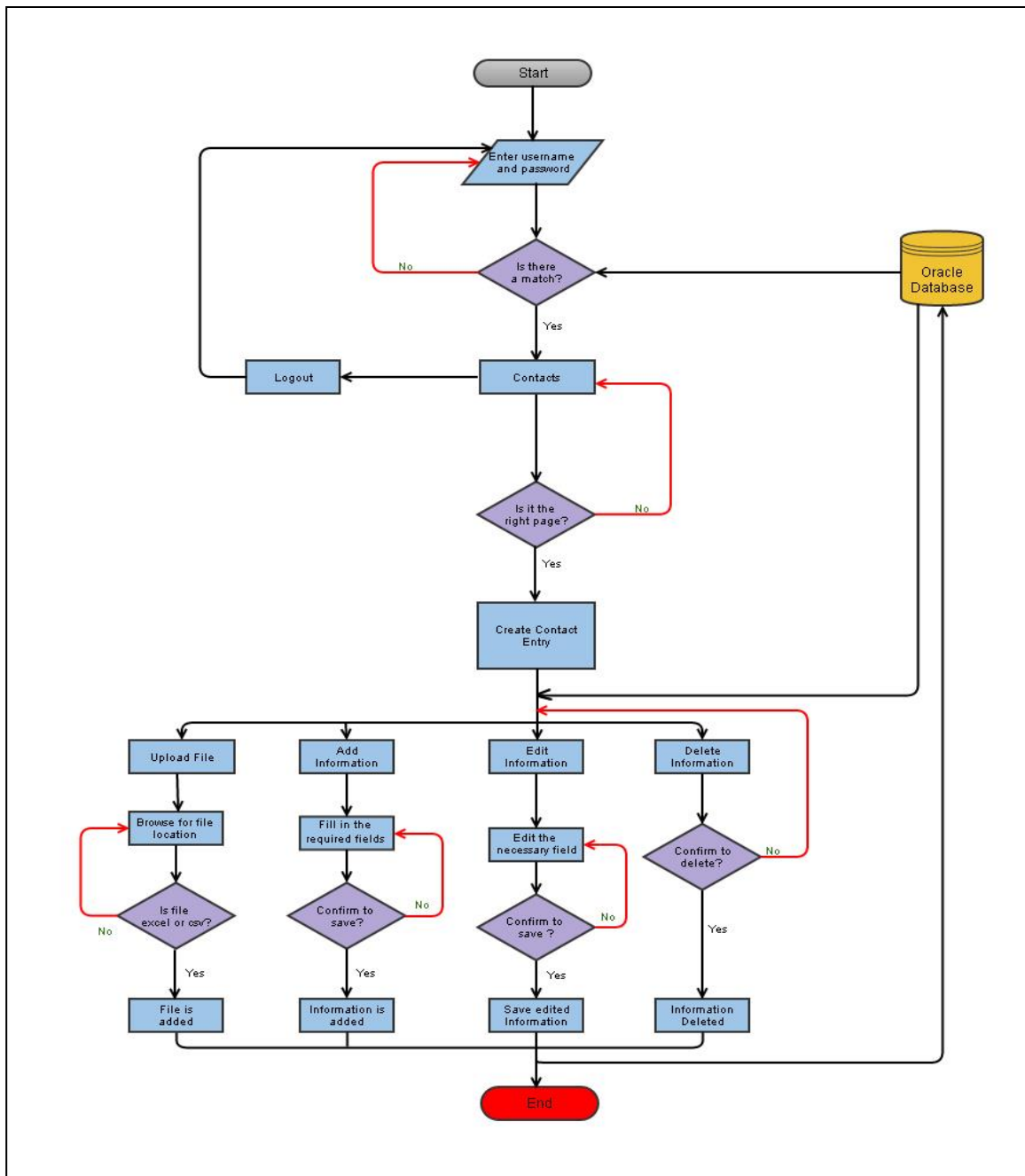


Figure 7. Flowchart of Contact Page

## Reports

The student report is made up of a summary of some pages with donut charts and Gantt chart which will graphically display some important records.

- Project Timeline: A group of task bars which the student will enter based on the progress of a particular project.
- Summary: A PL/SQL customized region which is a summary of all the key features of the application in a snap shot.
- Files: Each student is allocated a minimal size in the workspace of APEX, therefore limited files can be uploaded and saved. The file page is linked to a Google drive, so in cases where a student wants to save larger files, the link sends them to Google but they need to have an account.
- All Courses Taken: The student will enter all courses taken during a quarter, this will help them to keep track of the requirements needed for graduation.
- Homework Report: The student can enter his/her required homework and homework scores, the data are shown in the donut chart on the homepage. The chart is based on the total average homework scores, this will help the student to know how they perform in a course.

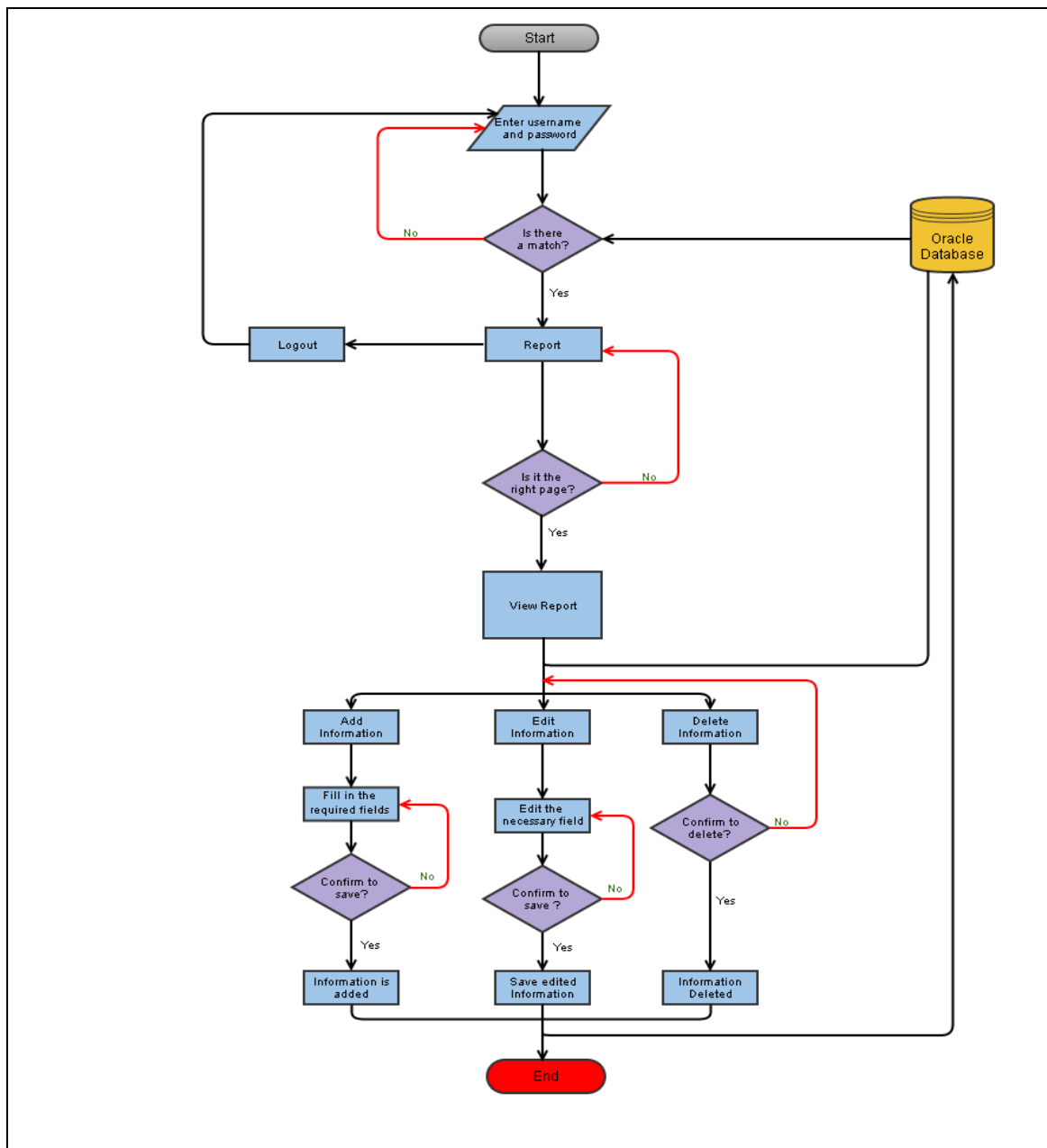


Figure 8. Flowchart of Report Page

## Notes

The student can create notes, each note has a timestamp to mark the time at which a note was updated.

- The student creates, edit and add notes. The note report shows the note name, the last updated, date created and descriptively
- The student adds notes
- The student can print notes
- The student can delete notes
- The student can make changes to note
- There is a last updated field which is a sysdate and gives the current time at which a note was updated

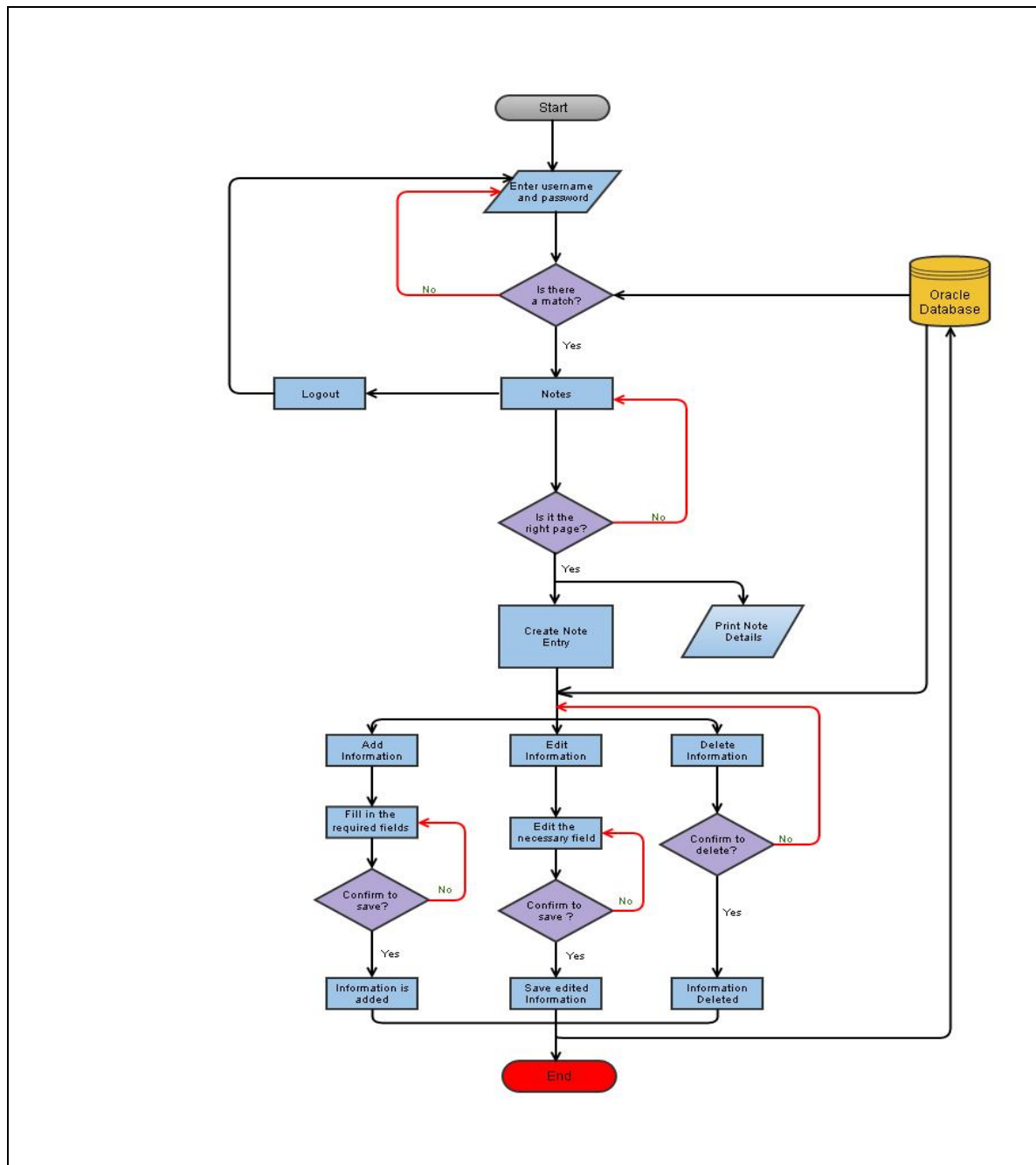


Figure 9. Flowchart of Note Page



## Events

The student will enter upcoming events and it will be displayed in a calendar, with the event name, date, time and location. The event is from the event table in the database.

- The student can add an event based on date and time
- The student can delete an event
- The student can make changes to an event
- The student can edit any field in the calendar and add colors to differentiate events

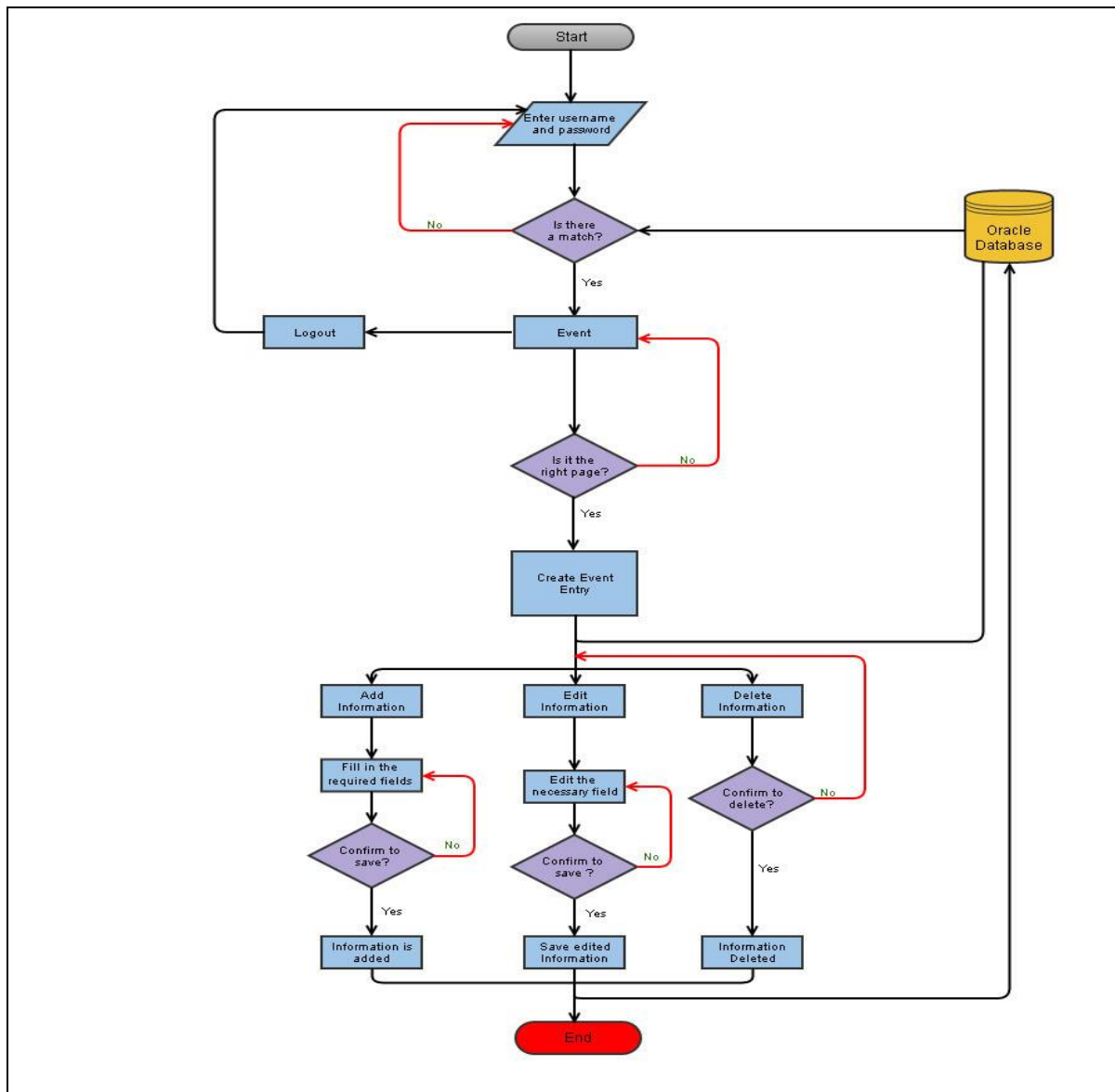


Figure 10. Flowchart of Event Page

## Resources

The resources tab will contain most of the basic important student information in the manual planner.

- **Maps:** The application has a map of the World, USA, California and the campus, for geographical purposes. They are very responsive and compatible with any mobile devices, with the ability to zoom in and out. The electronic campus map will replace the print out, where the locations are not eligible to read, in addition a student do not need to ask fellow students for directions, he/she just has to be registered to use the app and all the functionality of the map will be available to them.
- **Departments:** The link consists of all the departments in CSUSB with the room numbers, phone numbers and web address. The web address is links to their websites, this will make it easier for the students to search for any information they need for a department.
- **Important Dates:** A list of all the relevant dates of CSUSB, it includes holidays, deadlines for each quarter .
- **Other Campus:** This is a link to all the Cal State University in California with the programs they offer and their websites.

- Attractions and Restaurants: Attractions are fun places in California with their addresses, Restaurants are those within close proximity to CSUSB, specifically a mile, their addresses and phone numbers
- Writing Strategies: The various writing styles are defined here with some examples, this will be very informative to the student.

## High Level System Architecture Of Apex

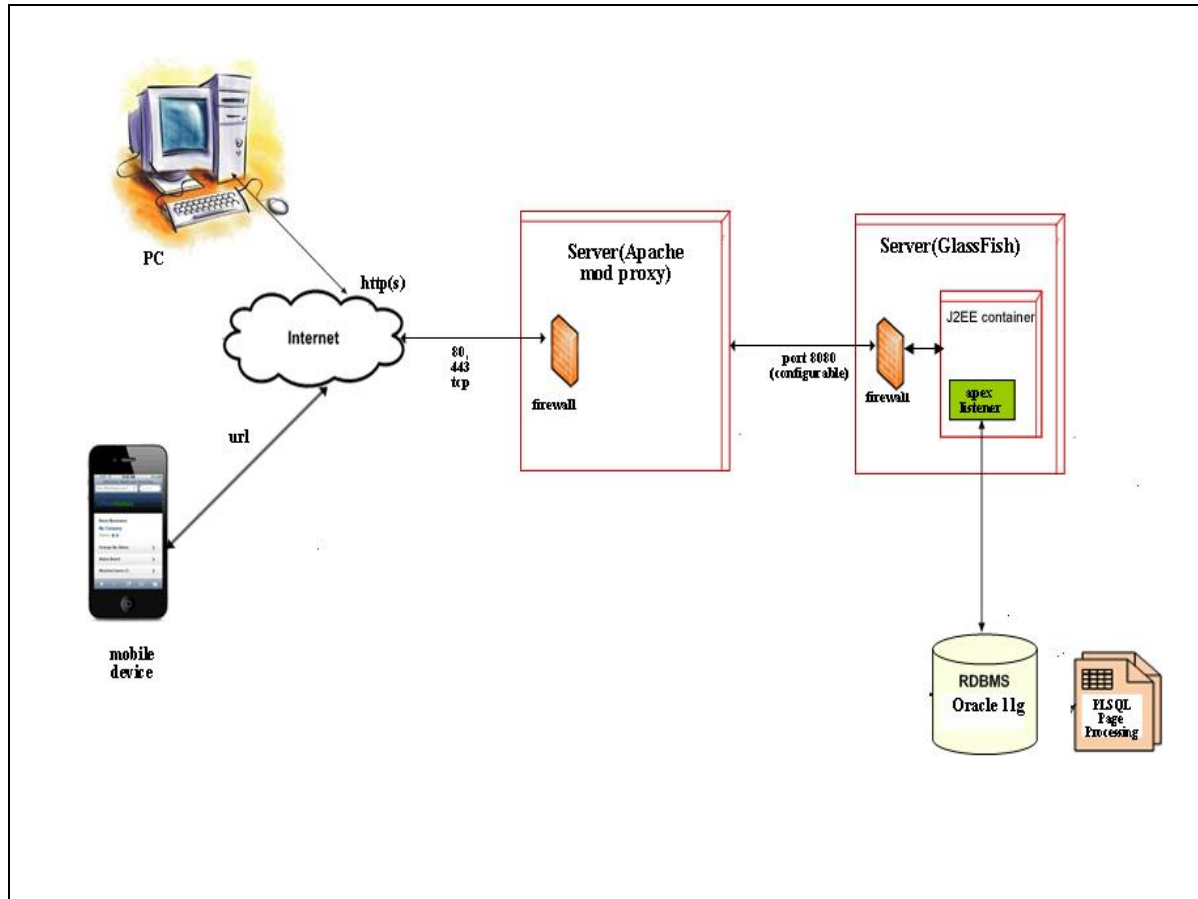


Figure 11. High Level System Architecture

The high level system architecture consists of an Apache web server, Glassfish web server, apex listener and the database. The APEX listener runs on any J2EE compliant application server. The Glassfish is one of the few servers supported by Oracle in terms of dealing with issues, so since the application is running on an Oracle database it is better to use that server application. The Apache web server is an option but will enhance security by adding another

firewall to the system, it can be installed on a different server from the other server, this will provide fault tolerance in a situation where the APEX application will be run on a large scale production. But it can be installed on the same server as the database and application server. The Apache server will help in situations where there is a use of certificates, only the Apache server needs the certificate for secure sockets layer connections because it is the only server that directly deals with the client web browser.

The web server sits between the web browser and the database and is responsible for handling the requests from the web browser, passing them through to the database (via something called apex listener), then APEX processes the request and generates the response (the HTML code to send back) which is passed to the browser via the apex listener.

The URL contacts web listener requests for a page, the web listener receives the HTTP request and parse it, looks at the path and maps it onto a call to run a PL/SQL procedure called "f". The URL of an APEX application is `http://machine:port/path/(f?page=)` , due to hacking and security issues checksum is required in session state protection to secure the URL.

The apex listener is a J2EE handler, it offers a few enhancements such as the ability to configure it via the web server itself, rather than via text files. Also, the student can define caching and security rules in the tool itself. Since the APEX Listener is a Java servlet it can be deployed using Oracle Web Logic

server (WLS), Oracle Glassfish, and OC4J. The firewall sits in between to allow the specified ports and protocols to allow access to the data, this makes data safe.

APEX is built on a declarative architecture, when the pages are created, the rows that correspond to the specific item on the page is inserted into the database, the same procedure will be executed over and over.

```
Command Prompt - sqlplus

SQL> select owner,object_type from dba_objects where object_name='f';
no rows selected

SQL> select owner,object_type from dba_objects where object_name='F';

OWNER                                OBJECT_TYPE
-----
PUBLIC                               SYNONYM
APEX_030200                          PROCEDURE
APEX_040200                          PROCEDURE

SQL> desc f
PROCEDURE f
Argument Name                        Type                                In/Out  Default?
-----
P                                     VARCHAR2                            IN      DEFAULT
P_SEP                               VARCHAR2                            IN      DEFAULT
P_TRACE                             VARCHAR2                            IN      DEFAULT
C                                     VARCHAR2                            IN      DEFAULT
PG_MIN_ROW                           VARCHAR2                            IN      DEFAULT
PG_MAX_ROWS                          VARCHAR2                            IN      DEFAULT
PG_ROWS_FETCHED                      VARCHAR2                            IN      DEFAULT
PSP_REGION_ID                       VARCHAR2                            IN      DEFAULT
SUCCESS_MSG                          VARCHAR2                            IN      DEFAULT
NOTIFICATION_MSG                    VARCHAR2                            IN      DEFAULT
CS                                    VARCHAR2                            IN      DEFAULT
S                                    VARCHAR2                            IN      DEFAULT
TZ                                    VARCHAR2                            IN      DEFAULT
P_LANG                              VARCHAR2                            IN      DEFAULT
P_TERRITORY                         VARCHAR2                            IN      DEFAULT

SQL> _
```

Figure 12. F-Procedure of Apex(Heart of Apex)



## Design and Implementation Constraints

### The Application Programming

- SQL, PL/SQL, JavaScript, CSS, AJAX, JQuery, HTML

### The Modules of the System Architecture

- Application Server(Glassfish or Apache Tomcat) and a web server (Apache)
- Oracle Database 11g ( 20g Storage Space)

An Oracle database is needed in order to create the application, but since the school already has Oracle 11g installed. The workspace will be created and the size specified, this will vary depending on the proposed number of student users. The developer will create the tablespace and specify which path the apex images will be stored. A schema will be created to hold the database and all its objects, the application is developed based on a schema.

During the APEX installation the required ports for the communication will be specified, this can be changed anytime by the administrator. The passwords can be set during this process.

During the installation of the Oracle Apex, certain parameters must be met, this will be an issue to anyone who wants to be a developer, but all the student needs is just a computer or smart phone to access the application.

If anyone is interested in installing oracle apex, he should find attached the step by step approach to installing apex 4.2. in the appendix B, all common errors have been addressed.

## Front-end Development

### Stand Alone (What Needs To Be Installed By The User)

All that is required to use an APEX application is a modern web browser, the student will enter the correct URL into the browser, and the relevant username and password. The theme for the system is very responsive on mobile phones and tablets. APEX applications can be as simple as a single page or contain multiple pages and interface with external systems via web services. The user will log in with a default username and password, and later customize his application with a changed password option. The student can use the system anywhere they travel, all the student needs is an internet connection. The application will work on any device irrespective of the Operating System unlike the native applications.

## Back-end Development

The system administrator creates the relevant fields, tables, performs all the tentative task to create the application.

### Web Service (Maintenance And Administration)

Consider whether to deploy the application to the cloud, or host on the school's database server. If a cloud implementation, all the Security, Performance, Scalability, Upgrades, Transaction control, Packaging, re-use and Web services will be taken care of by the provider as a service, where the client need to upload his application and pay for the services, but the prices are high. The Cloud Services by Oracle provides various choices, giving the option of a single schema, or a virtual machine with an Oracle Database instance. The Database Cloud Service also includes different storage sizes, administrative control and which application type to be used as the platform.

Due to the cost of deploying the application on the cloud, the application will be hosted on the school's Oracle database server, users will be created and given some of the administrative privileges, including how the user can get views of the database.

Administrative privileges belong to the super user of the application, the user has all the rights to the application. The administrator can create users, manage services, monitor various activities such as total number of people connected to the use the application, login attempts, workspace reports and other tasks.

For updates on the cloud as a host , when newer versions of Apex and the Apex code itself are introduced, the code is not out of date. The cloud maintains a compatibility of Apex application on a virtual machine, so the obsolete code will work fine without any changes . The Apex code runs when needed, hence the scalability, and maintenance are the responsibilities of the cloud service providers.

When hosted on the school's server, the administrator has the right to upload new codes and modification in an SQL file format, which can be sent through an email or can access the application with a team viewer through the permitted authenticated protocol. The administrator uploads updates only to the application server and become instantly available for all devices, all users will have the latest version of the application without having to run updates individually.

## Database Design

The backbone of APEX is the database, and since APEX is a proprietary development tool from Oracle, the database will be Oracle 11g. The Oracle installation comes with the Oracle Enterprise Manager, which gives a better graphical representation on how to manage the database. The GUI makes most of the functionality of the database very easy to use. The database will be managed with Enterprise Manager, SQL developer and the terminal or command prompt.

The database is a centralized database, for this reason there is a need for an advanced security setup which will be considered in chapter 4. The application will have a table of users which will hold all the users of the application, the usernames and their obfuscate passwords. There are some specific steps to take as an administrator to be able to use APEX for development purposes. These include

- Create tablespace
- Create a Schema
- Create tables
- Create relationships between tables

- Create Sequences (sequences are special objects in oracle db that are used to generate unique ids for the primary key field of the table)
- Create Trigger (populate the primary key value when a row is inserted; a procedural code that is automatically executed in response to certain events on a particular table in a database)

INSERT event, UPDATE event, DELETE event

- Install APEX , Setup Listener

Table 4: Data Dictionary of the Database Tables

| DATA DICTIONARY FOR INTERACTIVE STUDENT PLANNER APPLICATION |   |   |  |  |
|---|---|---|--|--|
| Entity  | Description   | Attribute   | Data type  | Attribute Description  |
| STUDENT   | An imaginary table, and will be the student who will be using the app.                              |   |  |  |
| COURSE  | A course offered at CSUSB   | COURSE_NBR<br>COURSE_NAME<br>QUARTER<br>NUMBER_OF_UNITS<br>TYPE   | VARCHAR2(13 BYTE) NOT NULL<br>VARCHAR2(30 BYTE) NOT NULL<br>VARCHAR2(20 BYTE)<br>VARCHAR2(20 BYTE)<br>VARCHAR2(20 BYTE)                        | Department name and course number<br>The name of the course name e.g. CSE610<br>The quarter a course is offered<br>Current total number of units for a particular course<br>This describes the type of course, for example Graduate, Undergraduate, General Education  |
| EVENT   | Event or workshop, a student plans to attend  | TITLE<br>START DATE<br>START TIME<br>END TIME<br>LOCATION<br>COMMENTS   | VARCHAR2(30 BYTE) NOT NULL<br>DATE<br>DECIMAL(3,1) NOT NULL<br>DECIMAL(3,1) NOT NULL<br>VARCHAR2(60 BYTE) NOT NULL                             | The title of the event or workshop<br>The date when the event or workshop will be held<br>The time when the event or workshop starts<br>The time when the event or workshop ends<br>Location of the event or workshop  |
| CONTACT   | This will be like a year book, where a student creates and adds a new contact                       | FIRST_NAME<br>LAST_NAME<br>GENDER<br>EMAIL<br>PHONE<br>PHOTO<br>- FILENAME<br>- MIMETYPE<br>- IMAGE_LAST_UPDATE | VARCHAR2(50 BYTE)<br>VARCHAR2(50 BYTE)<br>CHAR<br>VARCHAR2(100 BYTE)<br>NUMBER(10)<br>BLOB<br>VARCHAR2(400 BYTE)<br>VARCHAR2(225 BYTE)<br>DATE | The name identifies a contact<br><br>The image of a particular contact<br>Accepts the default name of the image<br>Determines resource that is requested by the web server, format: image/jpeg<br>The last date an image was updated   |
| STUDENT_COURSE  | A course actually taken by a student and chosen from the list of available courses offered at CSUSB | COURSE_NBR<br>QUARTER_YEAR  | VARCHAR2(13 BYTE) NOT NULL<br>VARCHAR2(25 BYTE)  | The course nbr from the course table, identifies courses offered at CSUSB, for example CSE460<br>The quarter/year a course is taken by the student   |
| HOMEWORK  | Work assigned in a class and must be submitted by the student                                       | HOMEWORK_NO<br>HOMEWORK_DESCRIPTION<br>DATE GIVEN<br>DUE DATE<br>TOTAL_POINTS                                   | INT NOT NULL<br>VARCHAR2(60 BYTE) NOT NULL<br>DATE<br>DATE<br>NUMBER NOT NULL  | The sequence number of the homework<br>Describes the subject content of the homework<br>The date on which a homework was given<br>The due date of the homework<br>Total score for the homework   |
| PROJECT   | Project assigned in a course and must be worked on by a student                                     | PROJECT_ID<br>PROJECT_NAME<br>DESCRIPTION<br>PROJECT_STATUS<br>GOAL<br>DUE DATE<br>TYPE                         | NUMBER NOT NULL<br>VARCHAR2(200 BYTE)<br>VARCHAR2(4000 BYTE)<br>VARCHAR2(10 BYTE)<br>VARCHAR2(4000 BYTE)<br>TIMESTAMP<br>VARCHAR2(20 BYTE)     | The sequence number of the project<br>The name of the project<br>A detailed description of the project<br>An indication of whether the project is of priority (hot, normal)<br>The objective or target of a particular project<br>The date by which a project is due<br>Indicates whether individual or team project |
| NOTE  | Student writes and save note for references   | NOTE_NAME<br>DESCRIPTION<br>DATE_CREATED  | VARCHAR2(45 BYTE) NOT NULL<br>VARCHAR2(4000 BYTE)<br>TIMESTAMP WITH LOCAL TIME ZONE  | The name of a note which will be saved<br>The content of the notes<br>Date and time at which the notes was created   |

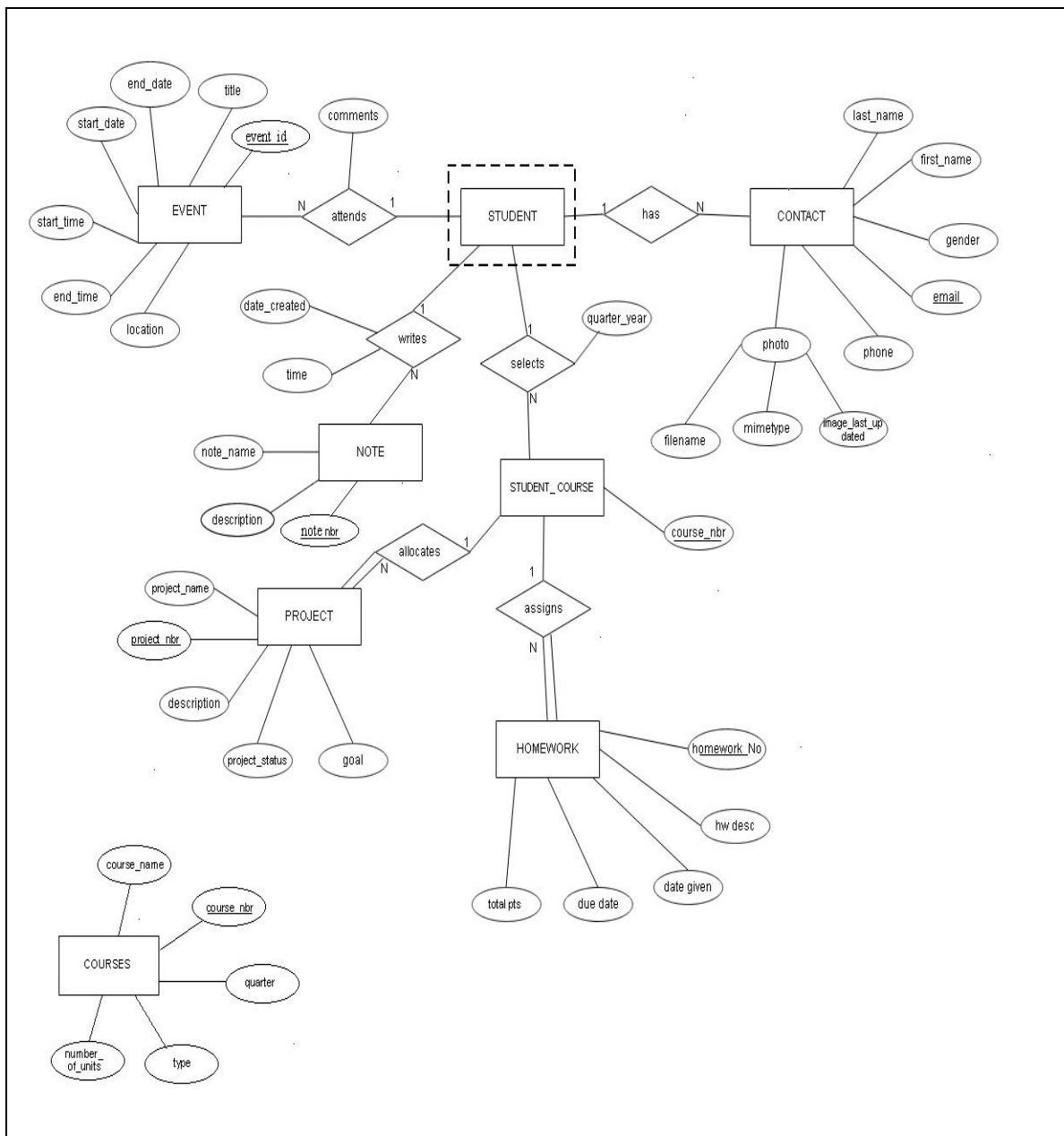


Figure 13. Entity Relationship Diagram



The Entity Relationship Diagram shown in Figure 13, is the logic and information flow of the system. The courses table is a lookup table for the courses offered in the Computer Science Department of CSUSB , there is no relationship between the course table and the other tables. Figure 14 is the schema diagram which shows relationships between tables, the dotted line shows an optional relation which indicates that there is one or no relationship.

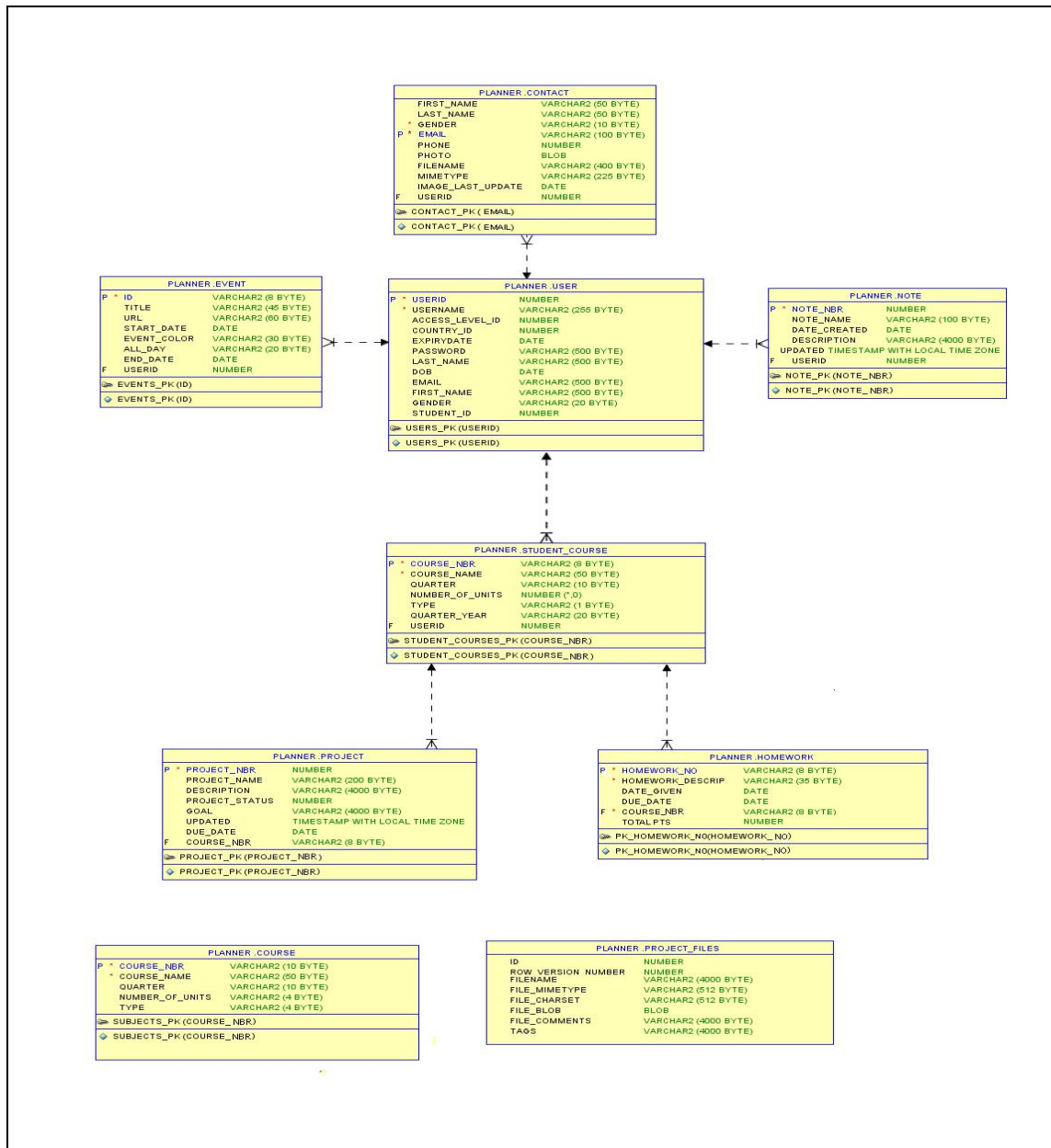


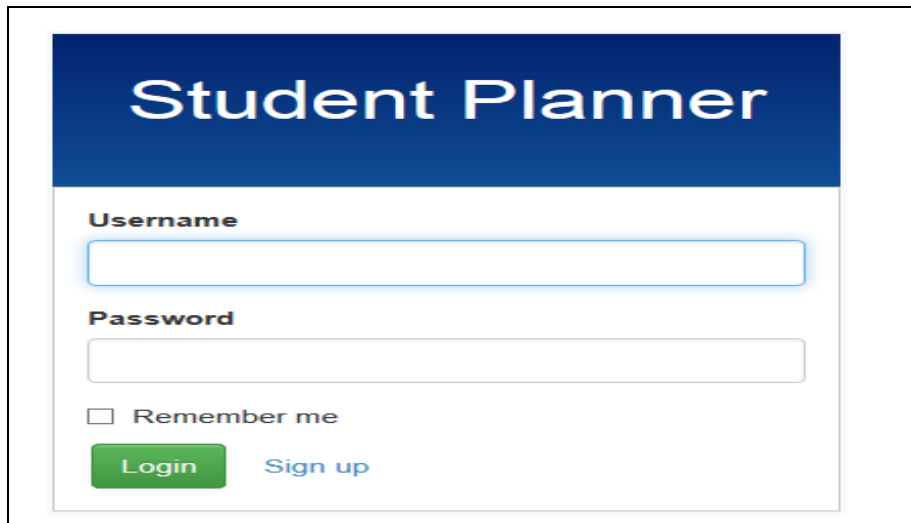
Figure 14. Unified Modeling Language Diagram

## User Interface

The user interface will be easy to use and very interactive so that the average computer knowledge person will be able to access. The navigation bars and tabs make it easy to move from page to page, in this section the design of each page is described, including : the components, region, buttons, items, computation and processes. A brief explanation of how the row fetch works and data manipulation is highlighted in the subsequent pages.

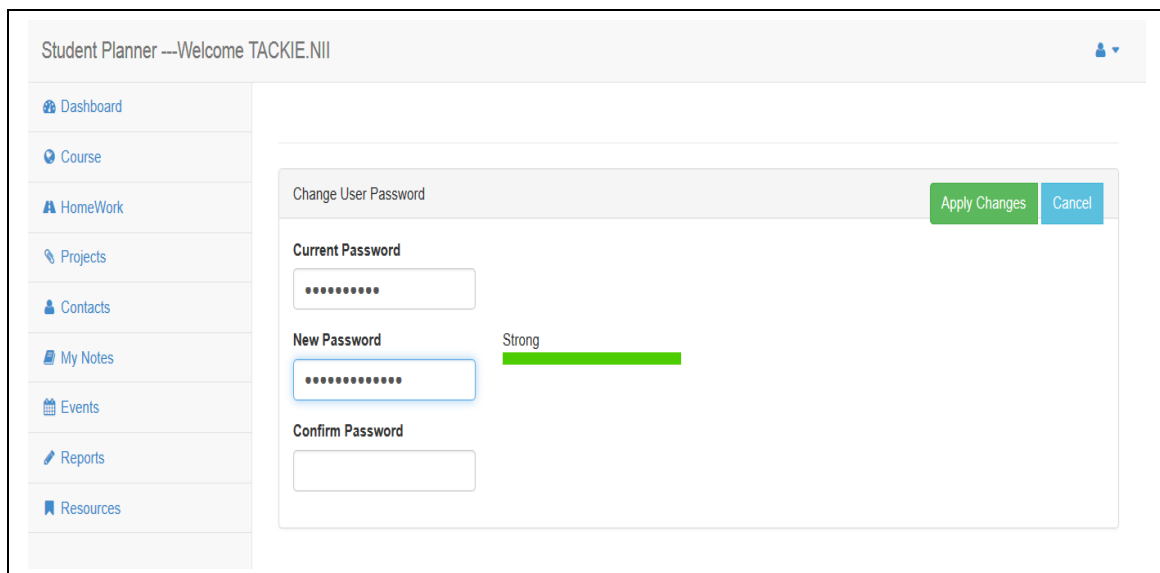
A first time user will see a login page where if they are registered will input the given username and password. Once logged in, the user can change the password to what they prefer. The password has a password meter bar to depict the strength of the password a user creates, in order to change the password, the current password must match the given password otherwise errors will be displayed and the new password must be the same as the confirm password. There is a profile page where the user can specify their country, email address, phone number and other biographic information. The user icon shows where the user can click to navigate to his/her profile page, change password or log out.

## Login User Interface and Change of Password



The image shows a login interface for 'Student Planner'. It features a dark blue header with the title 'Student Planner' in white. Below the header, there are two input fields: 'Username' and 'Password'. The 'Username' field is highlighted with a blue glow. Below the password field, there is a checkbox labeled 'Remember me'. At the bottom, there are two buttons: a green 'Login' button and a blue 'Sign up' link.

Figure 15. User Login Screen



The image shows the 'Change User Password' page in the 'Student Planner' application. The page has a light gray header with the text 'Student Planner ---Welcome TACKIE.NII' and a user profile icon. On the left, there is a sidebar menu with links to 'Dashboard', 'Course', 'HomeWork', 'Projects', 'Contacts', 'My Notes', 'Events', 'Reports', and 'Resources'. The main content area is titled 'Change User Password' and contains three input fields: 'Current Password', 'New Password', and 'Confirm Password'. The 'New Password' field is highlighted with a blue glow. To the right of the 'New Password' field, there is a strength indicator showing 'Strong' with a green bar. At the top right of the form, there are two buttons: a green 'Apply Changes' button and a blue 'Cancel' button.

Figure 15a. Change Of Password Page

## Homepage

The homepage is a summary of the application in the dashboard, the design is based on a HTML, CSS and PL/SQL code with user defined attributes set to the relevant column size. All the features shown are linked to the appropriate page.

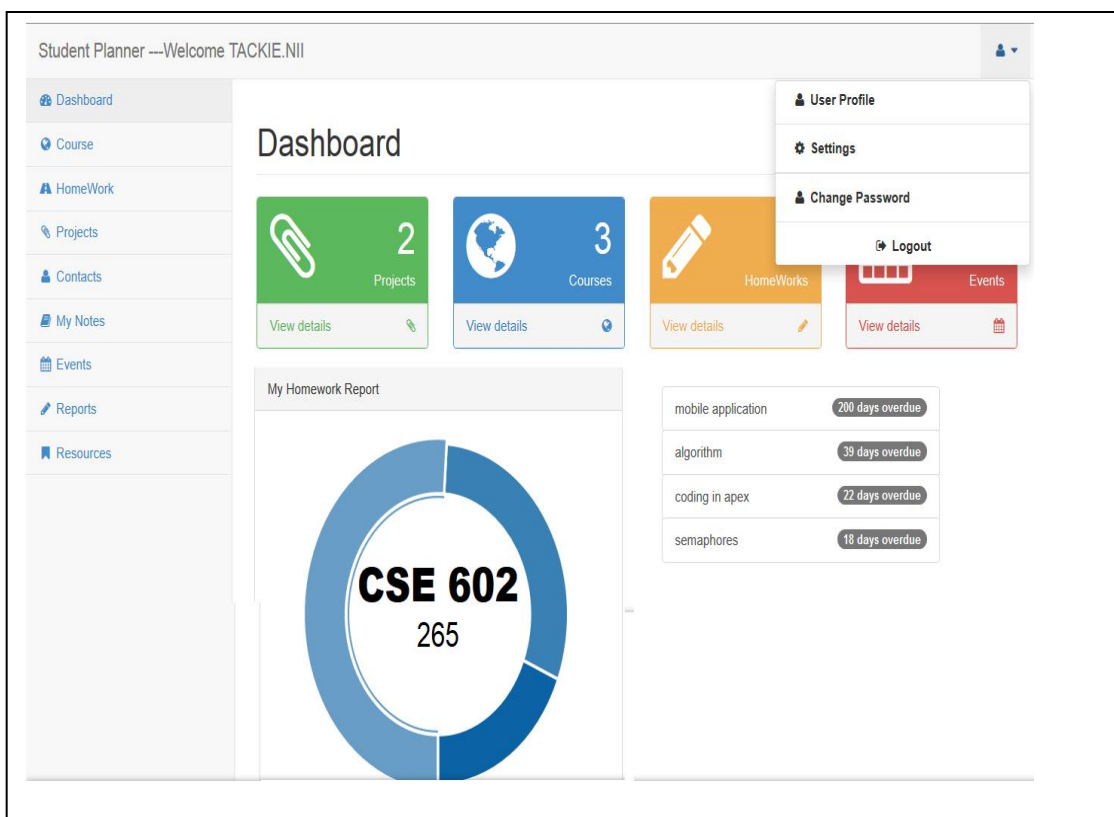


Figure 16: Desktop/Mobile Interface of Homepage

## Courses

The student can choose from a list of courses available to graduate or undergraduate studies or can enter the preferred courses.

Student Planner ---Welcome TACKIE.NII

Dashboard  
Course  
HomeWork  
Projects  
Contacts  
My Notes  
Events  
Reports  
Resources

Courses is a lookup for all the courses offered, it is grouped into several fields. Course cart holds all selected courses by the student.

Create Add Courses

Go Actions

| COURSE NAME                   | QUARTER | NUMBER OF UNITS | TYPE | SELECT                   | COURSE ID |
|-------------------------------|---------|-----------------|------|--------------------------|-----------|
| Bioinformatics                | FALL    | 4               | U    | <input type="checkbox"/> | CSE 122   |
| Programming in Visual Basic   | FALL    | 4               | U    | <input type="checkbox"/> | CSE 125   |
| Science,Computing and Society | FALL    | 4               | U    | <input type="checkbox"/> | CSE 129   |
| Computer Science I            | FALL    | 4               | U    | <input type="checkbox"/> | CSE 201   |
| Computer Science II           | FALL    | 4               | U    | <input type="checkbox"/> | CSE 202   |
| Object Oriented programming   | FALL    | 4               | U    | <input type="checkbox"/> | CSE 292   |
| Digital Logic                 | FALL    | 4               | U    | <input type="checkbox"/> | CSE 310   |

Figure 17: Desktop/Mobile Interface of Course Page with Lookup Table

Course Cart

Clear

| C001                            | COUNT |
|---------------------------------|-------|
| Formal Methods                  | 1     |
| Computation & Complexity Theory | 1     |
| Modern Comp.Architecture        | 1     |

Student Courses

Create

| Course Name                     | Quarter | Number Of Units | Type | Course Nbr |
|---------------------------------|---------|-----------------|------|------------|
| Formal Methods                  | WINTER  | 4               | G    | CSE 656    |
| Computation & Complexity Theory | WINTER  | 4               | G    | CSE 602    |
| Modern Comp.Architecture        | WINTER  | 4               | G    | CSE 610    |

Figure 17a: Desktop/Mobile Interface of Course Page

## Homework

The student keeps track of due dates of homework, and can edit all the fields. This is where the student enters the homework name, description, due date and some actions to be carried on.

The screenshot shows the 'Add Homework' form in a web application. On the left is a sidebar menu with options: Dashboard, Course, HomeWork (selected), Projects, Contacts, My Notes, Events, Reports, and Resources. The main content area has a green 'Saved Hwk' button at the top right. Below it is a message: 'Add Homework or go to saved homework to make changes to previously added ones.' The form itself has a title bar 'Add Homework' with 'Cancel' and 'Create' buttons. The fields are: 'Homework Description' (a large text area), 'Date Given' (a date input field with a calendar icon), 'Due Date' (a date input field with a calendar icon), and 'Course Nbr' (a text input field).

Figure 18: Desktop/Mobile Interface of Homework Page

The screenshot shows the 'Homework Report' interface. At the top, it says 'Student Planner ---Welcome TACKIE.NII'. The sidebar menu is the same as in Figure 18. The main content area has a 'Homework Report' title bar with a 'Create' button. Below the title bar is a search bar with a magnifying glass icon, a 'Go' button, and an 'Actions' button. The table below has the following data:

|  | HOMEWORK NBR | HOMEWORK DESCRIPT  | DATE GIVEN  | DUE DATE    | COURSE NBR |
|--|--------------|--------------------|-------------|-------------|------------|
|  | 42           | algorithm          | 22-SEP-2014 | 26-SEP-2014 | CSE 660    |
|  | 44           | semaphores         | 06-OCT-2014 | 17-OCT-2014 | CSE 660    |
|  | 46           | mobile application | 10-APR-2014 | 18-APR-2014 | CSE 602    |
|  | 81           | coding in apex     | 10-OCT-2014 | 13-OCT-2014 | CSE 602    |

At the bottom left of the table, it says '1 - 4'.

Figure 18a: Desktop/Mobile Interface of Homework Report

## Project

The student will track the progress of his project, print, upload, email and download files associated with a particular project.

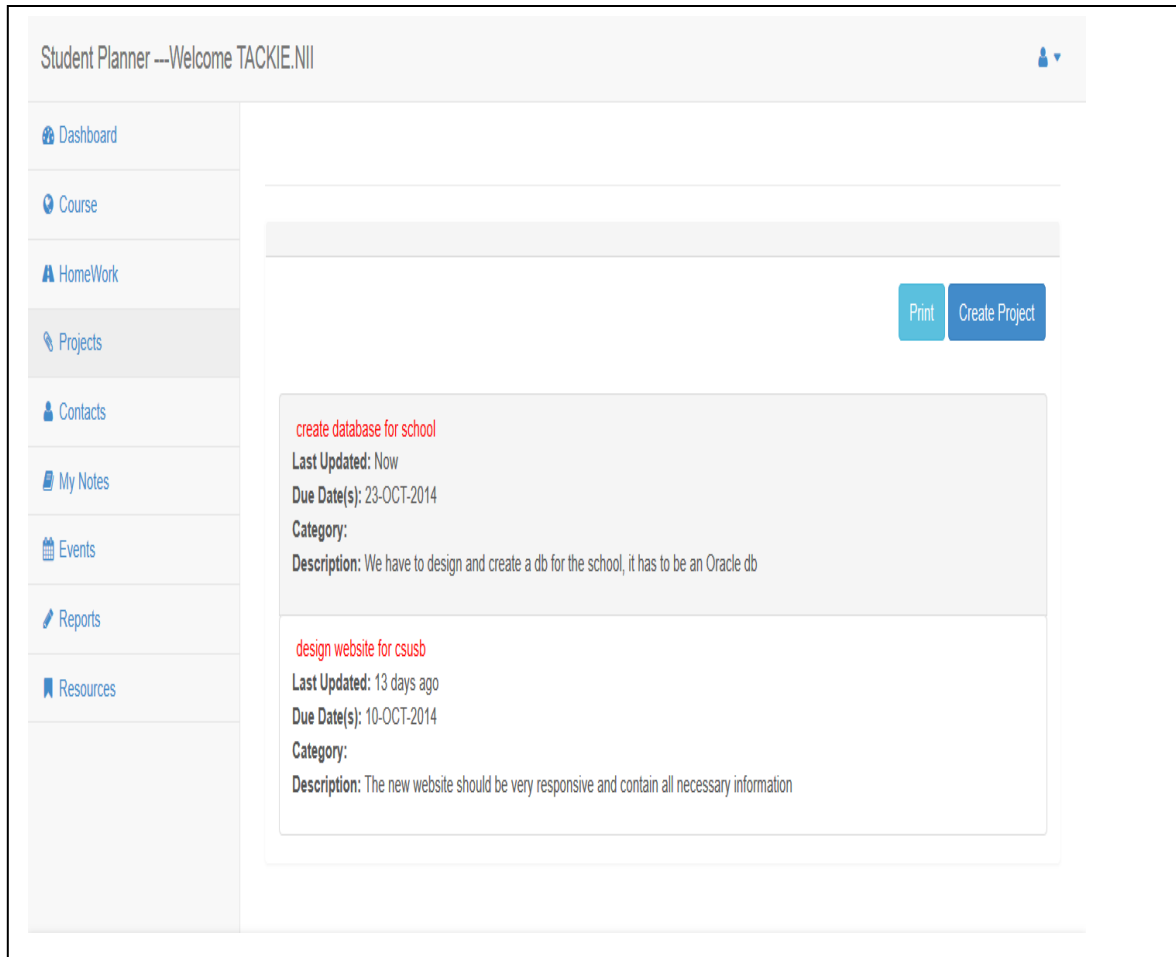


Figure 19: Desktop/Mobile Interface of Project Page



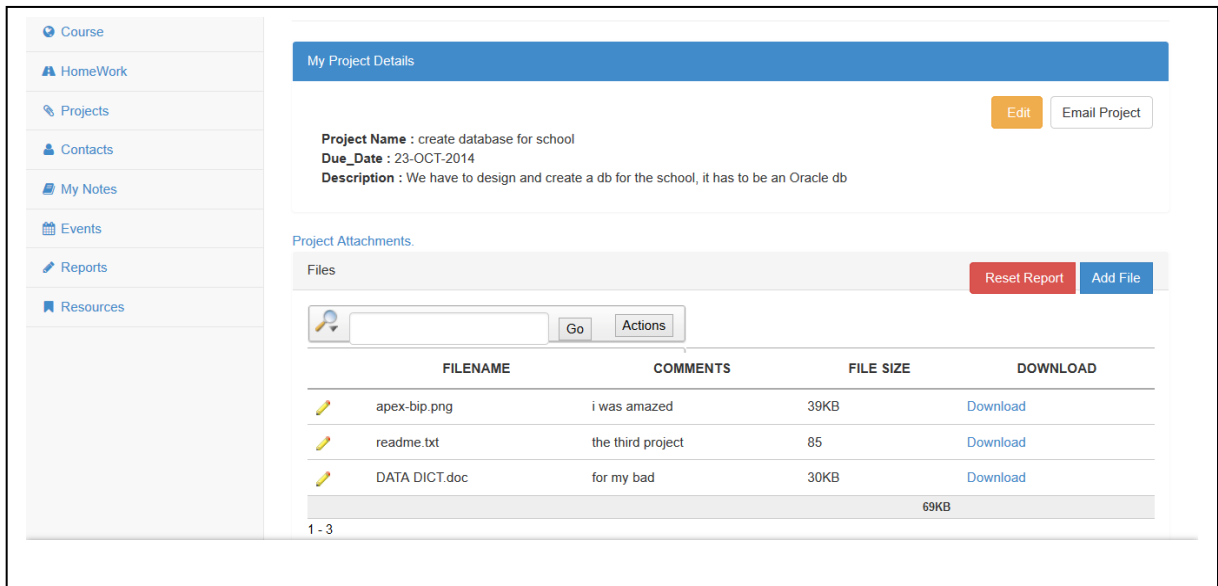


Figure 19a: Desktop/Mobile Interface of Project Details Page

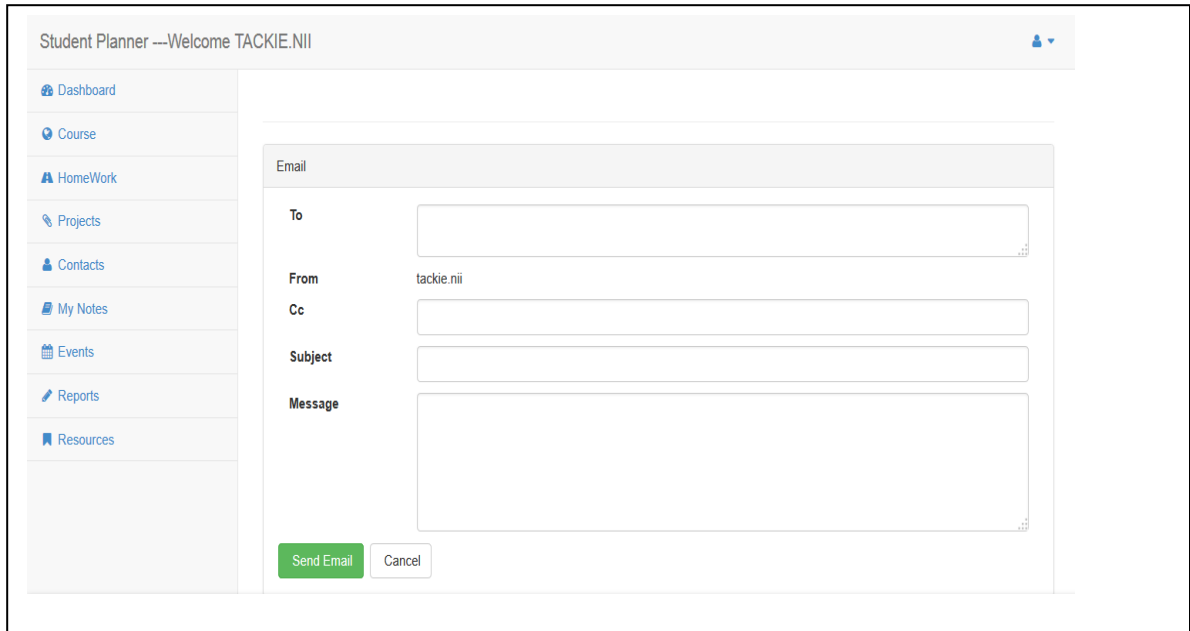


Figure 19b: Desktop/Mobile Interface of Send Email in Project Page

Project

**Project Name**

**Category** ☐ Group ☐ Individual

**Due Date**

**Description**

Cancel Add Project

Figure 19c: Desktop/Mobile Interface of Create Project Page

Student Planner ---Welcome TACKIE.NII

Project

Cancel Delete Apply Changes

**Project Name**  
create database for school


**Start Date**  
23-OCT-2014

**Description**  
We have to design and create a db for the school, it has to be an Oracle db

Figure 19d: Desktop/Mobile Interface of Edit Project Page

## Contacts

The student can easily update his contact by uploading the file in an excel form, or csv, hence no need to manually enter all contact details.




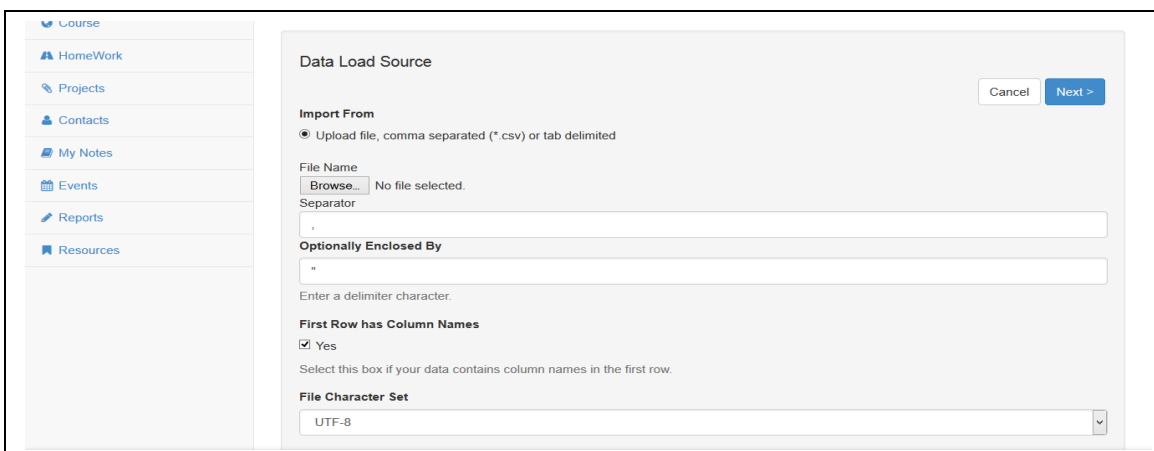
| FIRST NAME | LAST NAME | GENDER | EMAIL            | PHOTO  | PHONE |
|------------|-----------|--------|------------------|--|-------|
| jeremy     | Papafo    | M      | jeremy@yahoo.com |   | -     |
| Kwesi      | Koom      | M      | kk@yahoo.com     |   | -     |
| lebron     | james     | M      | ljames@yahoo.com |  | -     |

Figure 20: Desktop/Mobile Interface of Contact Page



**Data Load Source**

**Import From**

☒ Upload file, comma separated (\*.csv) or tab delimited

File Name:  No file selected.

Separator:

**Optionally Enclosed By**

Enter a delimiter character.

**First Row has Column Names**

☒ Yes

Select this box if your data contains column names in the first row.

**File Character Set**

Figure 20a: Desktop/Mobile Interface of Upload Contact

Student Planner ---Welcome TACKIE.NII

- Dashboard
- Course
- HomeWork
- Projects
- Contacts
- My Notes
- Events
- Reports
- Resources

Contacts Frm Cancel Create

**First Name**

**Last Name**

**Gender**  
 --Select Gender-- ▼

**Email**

**Photo**  
 Browse... No file selected.

Figure 20b: Desktop/Mobile interface of Add New Contact

Student Planner ---Welcome TACKIE.NII

- Dashboard
- Course
- HomeWork
- Projects
- Contacts
- My Notes
- Events
- Reports
- Resources

Contacts Frm Cancel Delete Apply Changes

**First Name**

**Last Name**

**Gender**  
 ▼

**Email**

**Photo**  
 Browse... No file selected.  
[Download](#)

Figure 20c: Desktop/Mobile Interface of Edit Contact

## Reports

The student report will be made up of a summary of some pages with pie charts which will graphically display some important records.

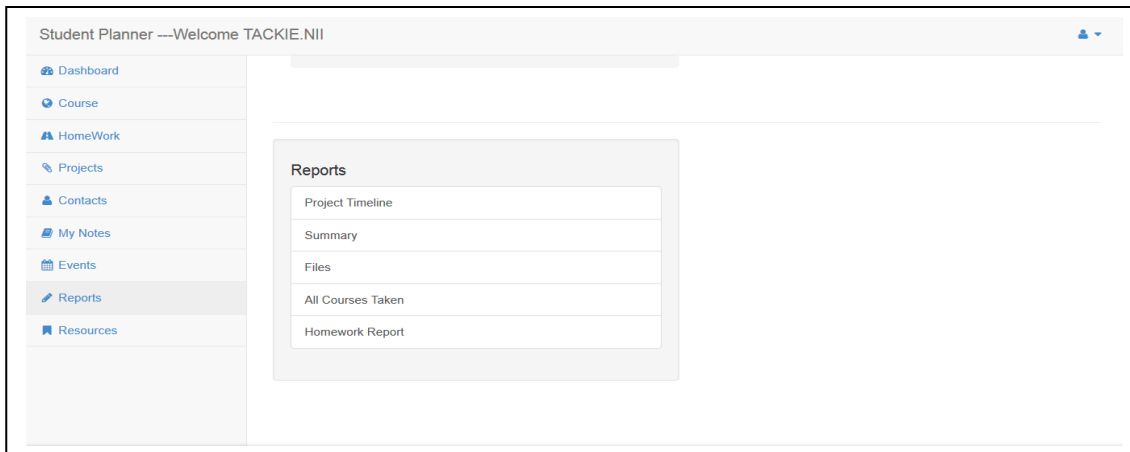


Figure 21: Desktop/Mobile Interface of Report Page

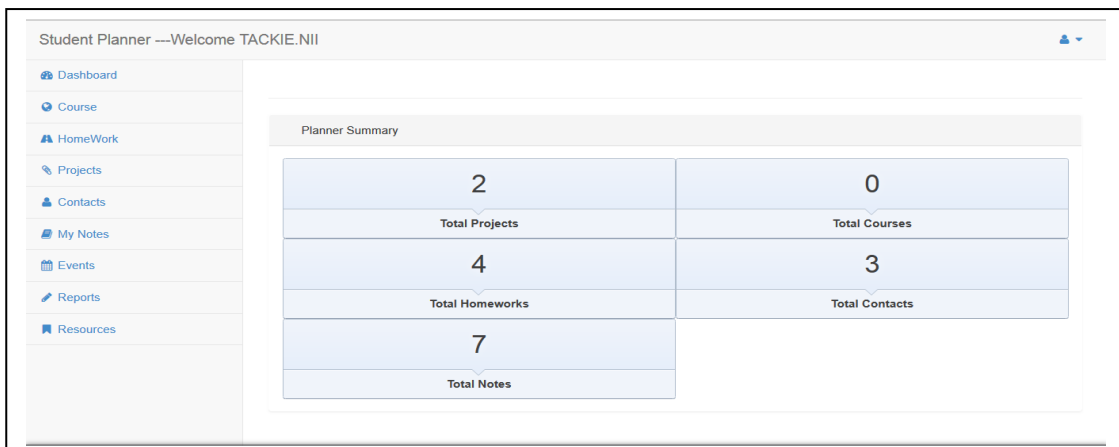


Figure 21a: Desktop/Mobile Interface of Summary Page

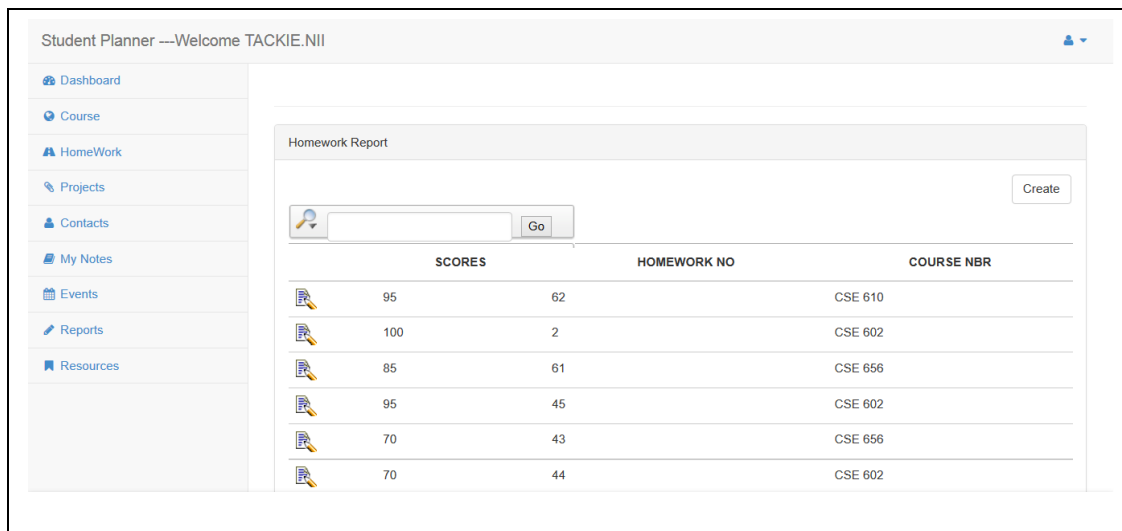


Figure 21b: Desktop/Mobile Interface of Homework Report

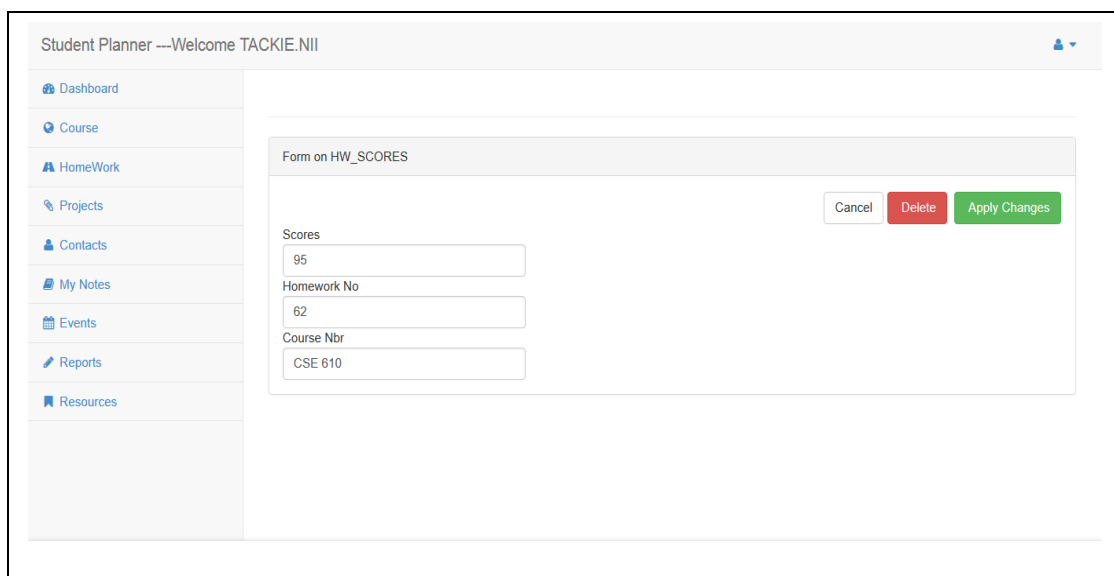


Figure 21c: Desktop/Mobile Interface of Homework Scores

## Notes

The student will create and edit notes, each has a timestamp to mark the time at which a note was updated.

The student creates, edit and add notes. It shows the note\_name, the last updated, date created and description.

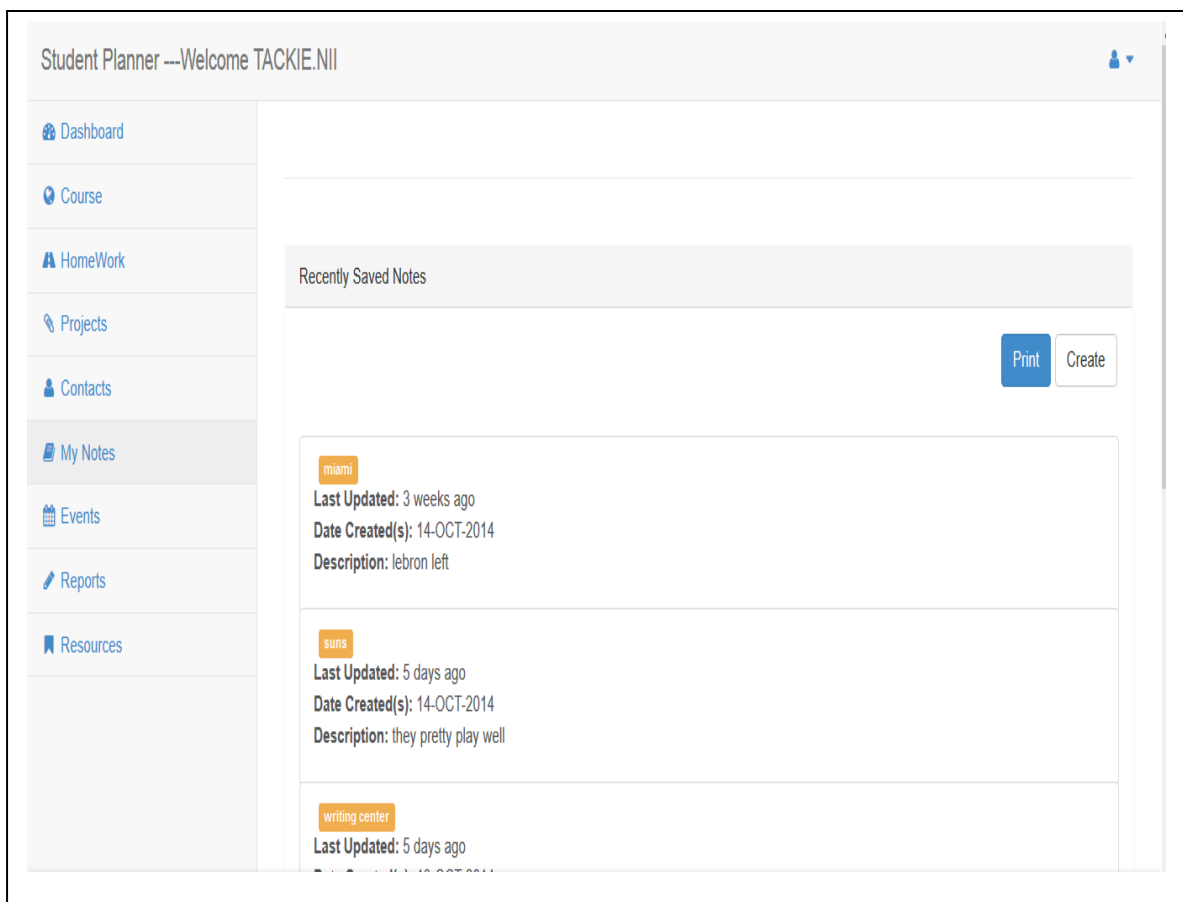


Figure 22: Desktop/Mobile Interface of Note Page

Dashboard  
Course  
HomeWork  
Projects  
Contacts  
My Notes  
Events  
Reports  
Resources

Create Notes Close Save Note

**Date**

**Note Title**

**Description**

**Screenshot/Supporting File**  
 No file selected.  
 Attachments must be under 15M in size.

Figure 22a: Desktop/Mobile Interface of Create Note

Course  
HomeWork  
Projects  
Contacts  
My Notes  
Events  
Reports  
Resources

Create Notes

**Date**

**Note Title**

**Description**

**Screenshot/Supporting File**  
 No file selected.  
 Attachments must be under 15M in size.

Close Apply Changes Remove My Note

Figure 22b: Desktop/Mobile Interface of Edit Note



## Events

The student will enter upcoming events and will be displayed in a calendar, with the event name, date, time and location. The student can easily edit or delete an event.

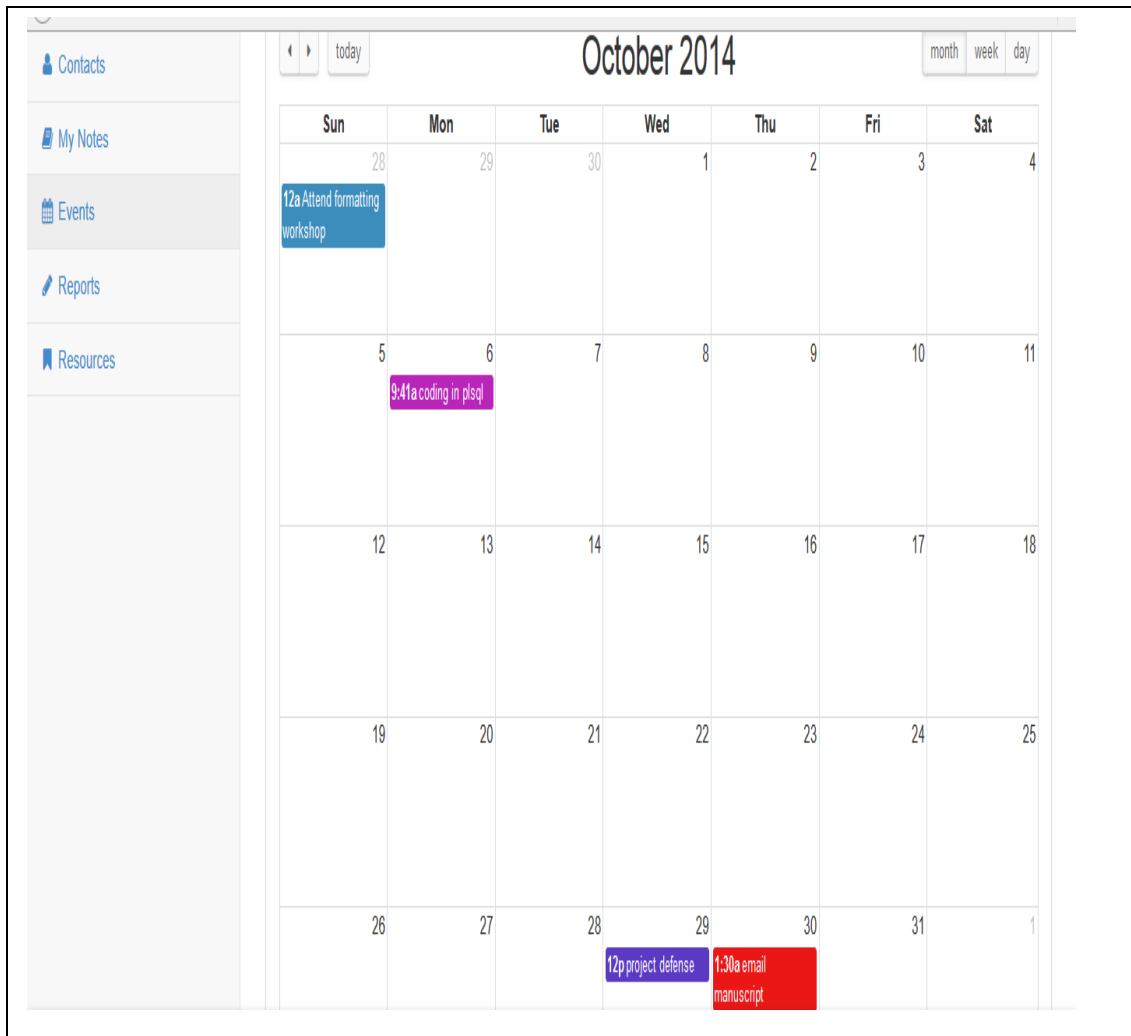


Figure 23: Desktop/Mobile Interface of Event Page

**EVENTS**

Cancel Create

**Title / Location**

**Url**

**Start Date**

**Event Color**

**All Day**

☐

**End Date**

Figure 23a: Desktop/Mobile Interface of Create Event

**EVENTS**

Cancel Delete Apply Changes

**Title / Location**

Attend formatting workshop

**Url**

Ontario

**Start Date**

28-SEP-2014 00:00

**Event Color**

#3c8dbc

**All Day**

☒

**End Date**

29-SEP-2014

Figure 23b: Desktop/Mobile Interface of Edit Event

## Resources

This page displays some relevant student information, the departments are linked to their websites, the campus map is very responsive on all devices and can be zoomed in to choose a location of interest.

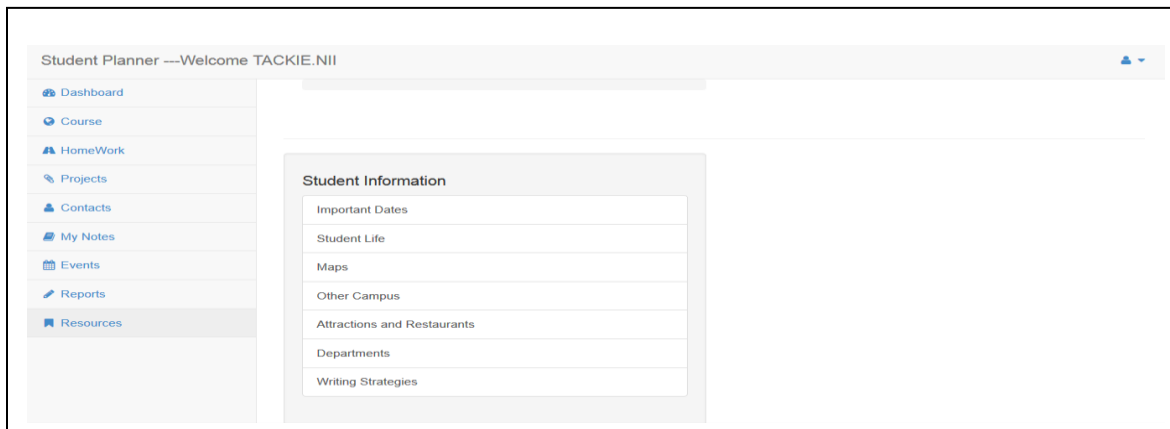


Figure 24: Desktop/Mobile Interface of Resources Page

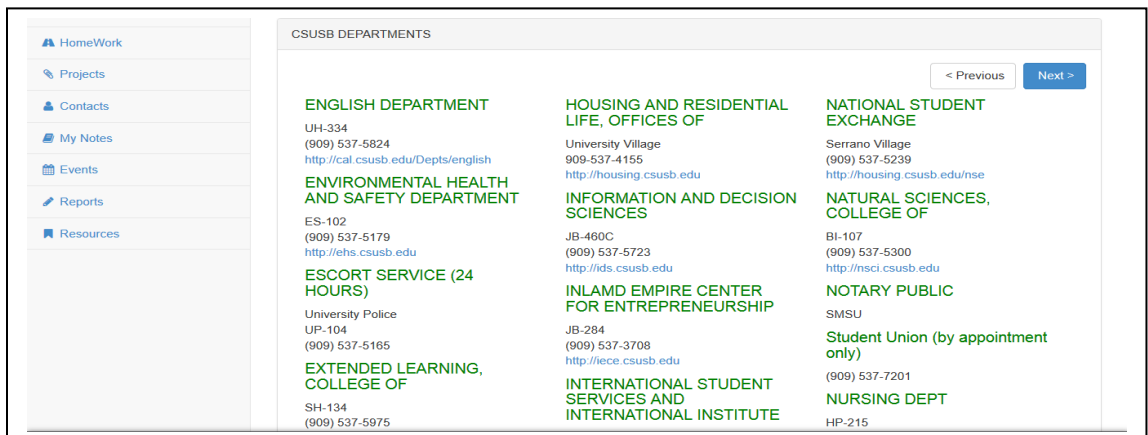


Figure 24a: Desktop/Mobile Interface of Departments in Resources Page

|           | Important Dates   |   |   |
|-----------|---|---|---|
|           | FALL 2014 QUARTER<br>(SEPT-DEC)                                       | WINTER 2015 QUARTER<br>(JAN-MAR)                                      | SPRING 2015 QUARTER<br>(APRIL-JUNE)                         |
| HomeWork  | 09/25 Fall classes begin  | 01/10 Winter Saturday classes begin                                   | 03/31 Cesar Chavez Holiday, Campus Closed                   |
| Projects  | 09/27 Fall Saturday classes begin                                     | 01/12 Winter classes begin  | 04/04 Spring Saturday classes begin                         |
| Contacts  | 10/01 Last day to add open classes on MyCoyote                        | 01/16 Last day to add open classes on MyCoyote                        | 04/06 Spring classes begin                                  |
| My Notes  | 10/01 Census: Last day to drop without record of enrollment           | 01/19 Martin Luther King Jr. Holiday, Campus Closed                   | 04/10 Last day to add open classes on MyCoyote              |
| Events    | 10/27 Winter 2014 Advising begins                                     | 02/02 Census: Last day to drop without record of enrollment           | 04/24 Census: Last day to drop without record of enrollment |
| Reports   | 11/03 - 11/25 Winter 2014 Priority Registration                       | 02/09 Spring 2014 Advising begins                                     | 04/27 Summer 2014 Advising begins                           |
| Resources | 11/11 Veteran's Day, Campus Closed                                    | 02/16 - 03/09 Spring 2014 Priority Registration                       | 05/04 - 05/20 Summer 2014 Priority Registration             |
|           | 12/06 Last day of Fall Saturday classes                               | 03/10 - 03/13 Spring Open Enrollment (pre-payment required)           | 05/11 Fall 2014 Advising Begins                             |
|           | 12/01 - 12/05 Winter 2014 Open Enrollment (pre-payment required)      | 03/21 Last day of Winter Saturday classes                             | 05/25 Memorial day, Campus Closed                           |
|           | 11/27 - 11/30 Thanksgiving Holiday, Campus Closed                     | 03/23 Last day of Winter regular classes                              | 05/21 - 06/12 Fall 2014 Priority Registration               |
|           | 12/05 Last day of Fall regular classes                                | 03/24 - 03/28 Winter Finals Week                                      | 06/13 Last day of Spring Saturday classes                   |
|           | 12/08 - 12/13 Fall Finals Week  | 03/23 Spring 2014 Late Registration (late fee & pre-payment required) | 06/15 Last day of Spring regular classes                    |
|           | 12/15 Winter 2014 Late Registration (late fee & pre-payment required) | 03/30 - 04/03 Spring Break, Campus Closed                             | 06/16 - 06/20 Spring Finals Week                            |
|           | 12/16 Fall grades available on MyCoyote                               |   |   |
|           | 12/24 - 01/04/15 Winter Break, Campus Closed                          |   |   |

Figure 24b: Desktop/Mobile Interface of Important Dates



Figure 24c: Desktop/Mobile Interface of Campus Map

## CHAPTER FOUR

### DATA SECURITY

#### Introduction

This is the most important part of the application, issues pertaining to threats will be addressed , the causes and solutions will be provided, with some examples. There are three main security issues with the Interactive Student Planner application, securing the network, securing the application and securing the database. Much emphasis will be placed on the latter two security issues with the student planner. Application security and database security, but unlike the manual systems where an unauthorized person can get easy access to a student handout planner. The system ensures that the right security schemas are put in place.

The real power of APEX is based on the fact that most processing is done where the data resides, in the Oracle database. This demands that more complex setup is used for the application developed in APEX in order to protect data from intruders. The Interactive Student Planner runs on a database application hence securing the data is very valuable and of highest priority to the end user. The database security will entail some range of information controls to allow authorized users access to databases and its data. An unauthorized user must not be able to get access to the data, make any changes to the data, or the

data stored functions of the student planner application. The system covers various types of controls, such as technical, administrative and physical.

The diagram below is retrieved from Microsoft Patterns & Practices, it shows the various ways and levels of improving web application security. The student planner application follows similar 3-tier architecture and layers to ensure security which will help reduce the likelihood of an attack and mitigate the extent of damage if the attack occurs.

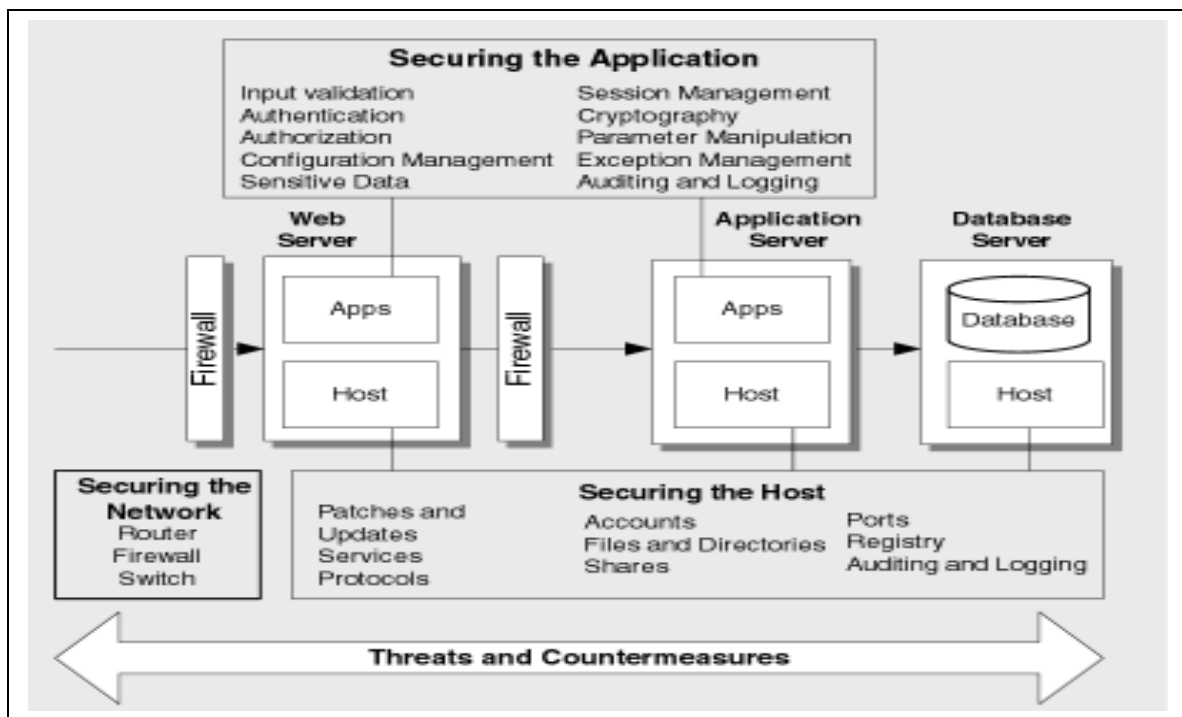


Figure 25. Scope of Improving Web Application Security:  
Threats and Countermeasures

Retrieved from <http://msdn.microsoft.com/en-us/library/ff649874.aspx>

## Types of Data Security

The two main types of security is addressed here, Application Security and Database Security .

### Application Security

URL tampering. The URL for APEX is in the `http://machine:port/path/(f?page=)` : local host of the machine on which APEX is installed, the port is the configured port to which APEX pages communicate between the web browser and the database, the path of where the file is located and the page identifier.

Hacking or tampering with such a URL is very easy and obvious because it clearly shows the page number which can be changed. Any user can easily manipulate the URL and get access to unauthorized page.

There are several ways to deal with this issue in APEX, the more secured is the use of the session state management, this prevent editing of the item value in the URL. The system session state will be configure in the APEX application during development , the page access prevention, application access prevention, and the arguments are set to have a checksum. The added checksum helps in a way that before the page is rendered , the APEX engine compares the checksum in the URL to a precompiled value, If there is any difference, APEX will not display the page but show an error message.

Authentication. The process of identifying each user before they access the application, it may require the user to enter a username and password or the use of digital certificates or a secure key. The system has a function in the database which references the username and password in the users table, a call to this function will be the authentication of the application. The created hash function will hash the password from prying eyes on the database. Once the user enters the system with the username and the password, a change of password screen appears which allows them to reset their password. When all is done the customization of the application is finished.

Authorization. This will help protect applications, pages, application components and extend. Authorization goes beyond simple user authentication, it helps with more options control access to the application.

Oracle APEX allows to create users and use the function available in APEX to restrict access to certain areas of the application.

Username:

Password:

Email Address:

Default Schema:

User is a developer:



User is a workspace Administrator:

The system consist of access\_level table which is only accessed by the administrator, the various levels of controls are set on the table and a PL/SQL function in the application returns the set access\_level based on which user has that privilege.

Page Flow Control. The system has an access control page which will define which users can have access to which part of the application. the access control list consist of view, edit and administrator privileges. An authorization scheme is created to manage access by user privileges.

#### User Privileges

- Edit: the student can edit the application but will not be able to make changes to any user access
- View: the student can only view the information, but cannot make any changes to the page.
- Administrator can change anything in addition to changing the user privileges
- Certain areas of the application will be restricted

Validations Processing. Apex allows the developer to set various validations levels to a page, it could be an item on the page or the whole page. The system will specify the location where an error message is displayed if the validation fails. The inline with field and in notification error display page with a PL/SQL validation type is used in the system. This makes sure that all or some specified conditions are met on an item or the page.

User. Access control will be the core concepts and first line of defense, this will limit each user to the application, since the application requires the user to interact more with the objects of the database such as tables, columns and SQL objects, views and stored functions.

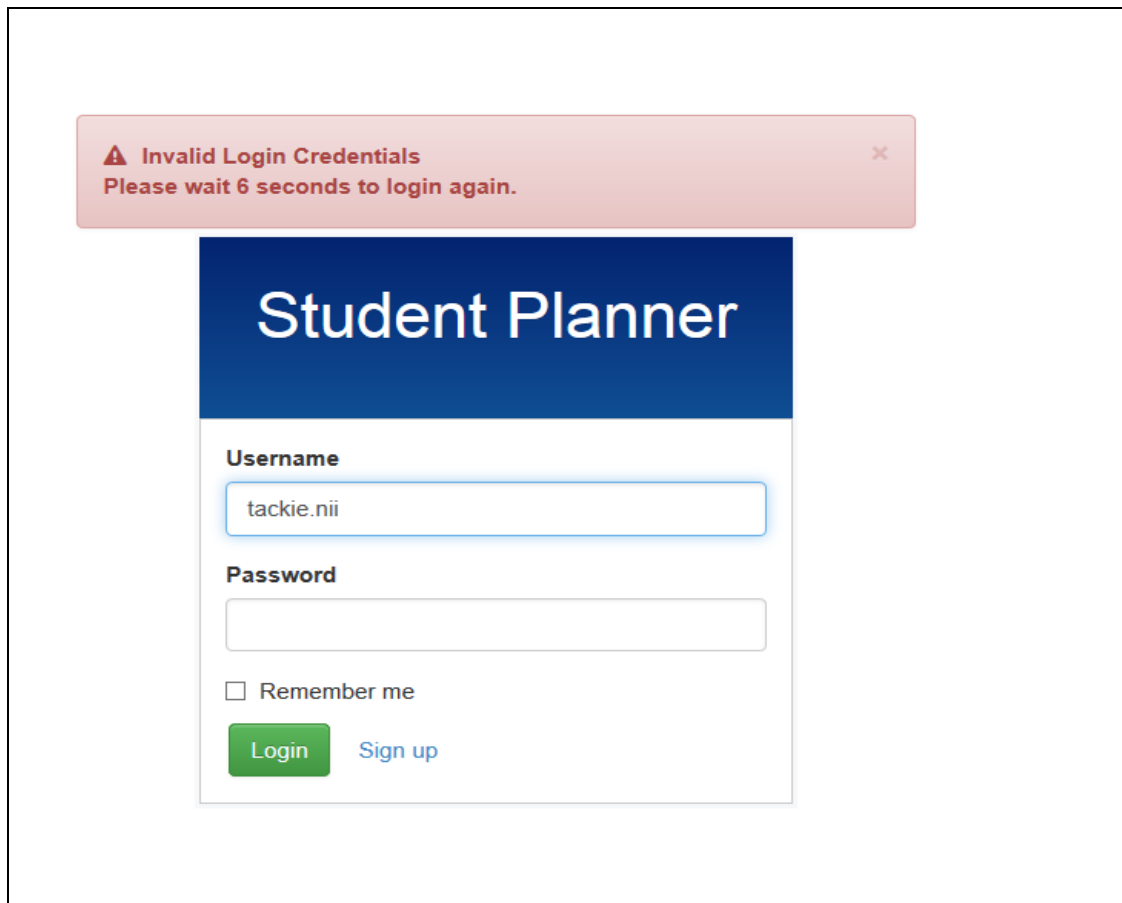
The user may perform DML operations, insert, select, insert, update and delete. These actions have a serious effect, so the user must be responsible for what they do.

Some part of the application allow for read privileges, select only, update only.

With the access controls, the system uses login credentials, the login will contain a default username and password. Once the student login to the system, the user can change the username and password to customize what the user want. The user can logout, once he has finished using his access password protected.

The system will log the student in only after identifying and authenticating the student credentials that the student inputs.

An error message is displayed when the login credentials do not match the required type, [e.g., (see Figure 26)].



The screenshot displays the 'Student Planner' login interface. At the top, a red error banner with a warning icon and a close button (X) reads: 'Invalid Login Credentials Please wait 6 seconds to login again.' Below this, the 'Student Planner' title is centered in a dark blue header. The login form includes a 'Username' field with the text 'tackie.nii', a 'Password' field, a 'Remember me' checkbox, and two buttons: a green 'Login' button and a blue 'Sign up' link.

Figure 26: User Access Control Login Page

### Server(Glassfish)

The more complex and high level setup of APEX will be used, the system is deployed with an Apache server in the front which will be accessed through specific ports. Then another separate server that runs Glassfish, the APEX listener will be deployed on the Glassfish, the APEX listener will access the database. Firewalls will be in between to allow traffic through the specific ports and protocols, this will make data safe from intruders .

The Glassfish Server is a middle wear server that facilitates easy information flow and security, it provides tools for managing security. The administration console is a browser-based utility that is used to configure security for the server. This includes users, managing certificates, groups, and realms. It has a wide range of system securities: Authentication, Authorization, Auditing, Firewalls, Certificates and SSL, Tools for Managing System Security.

The system will use the firewall because the Authentication, Authorization, Auditing will be done from the database side, the firewall will control the flow of data between different networks, and manage the network connections. The firewall is configured so that the user can access the required TCP/IP ports, assuming the HTTP listener is operating on port 8080, the system firewall allows HTTP requests on port 8080 only

## Database Security

The database will be a centralized database, because it will be inefficient to create a database for all the student. All SQL and PL/SQL commands issued by the application will be performed with the privileges and rights of the database schema defined below

Auditing. This is usually set up by the database administrator to track how the database is being accessed and the activities of the user. With auditing the database administrator can know what actions were performed, what data was changed, who accessed the database objects. Auditing does not prevent breach, but helps to find if there is any occurrence of breaches.

The audit helps monitor access attempt such as retaining information on successful and unsuccessful logon and log off attempt, records changes by the user and role privileges, user additions, user deletion, changes to the database schema such as changes to table structure or attribute data types. The auditing is implemented by log files and audit files.

Row Level Security. Row level security is important in this application because data related to many different users are stored in the database. It would be wasteful to assign each student to their own database, hence the need of a centralized database , but we cannot give full access to the data by the student. Consider an instance where a student should only be able to view or modify the rows of data that matters specifically to him/her. We can achieve this through creating a row level security or virtual private database.

Row level Security or Virtual Private Database (VPD) enables the developer to enforce security, directly on tables, views or synonyms. It provides row level access control beyond that of views, since it uses all the data manipulation languages transaction types, INSERT, SELECT, UPDATE, DELETE.

The mechanisms enable different users to work on the same schema, the access control is such that each user will feel as if there are accessing their own virtual private database. When two or more users access the centralized database in a web hosting environment or in the APEX application the VPD makes sure that each user will only see their data, as if each user has its own database but there is only one database.

VPD works by associating one or more security policies with tables of views, access to a table with an attach security policies causes the databases to consult a function , that returns a predicate which the database attach to the users SQL query statements. These policies enforce access condition which are appended

to user's SQL query statements dynamically modifies the data access. VPD can be implemented using stored procedures

The implementation of the where clause in the validations and process stage also helps in row level security. The "where" clause below can be included in the conditional statement of the SQL

where userid =( select userid from users where username = :APP\_USER)

SQL Injection. The user can influence the SQL code, this is usually as a result of the developer not properly using the right code. The user with some knowledge of SQL will like to put in some values in the form to try and query a database, this might select information from APEX application item, find items, and learn about the application and then can hack the app easily, or execute any DDL command that will alter the database objects.

Situation where the SQL statement is concatenated by substitution strings and static portion of the query, such that the user provides the values.

A section of PL/SQL that is susceptible to SQL injection

```
l_sql := 'SELECT *FROM project
```

```
WHERE project_name = '' ||p_project_name|| ''';
```

When a user enters any valid name in the database report, but not the right data, the database only ensures that the statement is syntactically correct , and parses it. Some records can be obtained instead of the required one.

Solution to this attack is to use a bind variable, these are evaluated during the bind phase of processing a query, the SQL will be processed, the strings are built in the bind variables. SQL injection cannot affect it.

A section the code with a bind variable

```
l_sql:= 'SELECT * FROM project  
  
        WHERE project_name =:project name;
```

The SQL is no more concatenated, the bind variable used is :project\_name when binding in the value of the item, this will make sure that whatever value is entered is the exact that in the table and raises an error if not found.

Dynamic SQL could be used as well where it is made up of a static compound and variable component, this will return records that match the exact value the user enters.



Cross- Site Scripting (XSS). The system expects the users to input expected data, but they do not, some advanced knowledge of JavaScript is required to execute such an attack.

APEX allows developers to choose which characters are allowed on an item, this escapes any data that contain bad tags. When there is a violation an error message will be generated.

Encryption. Applying an algorithm to encode messages or data with a key, such that only authorized users can read the data. The flow of data in any web application or APEX application is simple, the student sends data over a network to the web server, from the web server to the database, in APEX this request will be in the user's session, when there is a commit the data is written into the database files.

Data moves from one point to the other, during these stages, the data can be susceptible to attack, so an encryption mechanism has to be put in place to render the data being transmitted protected and invaluable to an intruder.

APEX allows some techniques to be implemented and configured to ensure the encryption is done well.

First the system considers the origin of the data, the web browser and the web server. HTTPS ensures that all data sent to the web server is encrypted. When an intruder tries to sniff, with the help of HTTPs the information obtained

will be useless and meaningless. The HTTPs will be configured on the web server , then enabled in the APEX application, when the HTTPs is enabled it will allow HTTPs for both outbound and inbound transactions.

The secure attribute in the session cookie attributes will also allow only application to run over HTTPs

Apex provides an item-level security attributes when enabled, stores session state values in an encryption format. This will be done on any item in the application, hence a more tedious approach but worthwhile. The option "store value encrypted in session state" when set to "Yes", allows APEX to store any value associated with that item encrypted.

DBMS\_CRYPTO package comes with the database, it contains APIs which can be used to both encrypt and decrypt data before stored in a the table. The developer will have the task of setting the API to be called anytime a request of an encrypted data is needed, set in various places of the application. The passwords column in the table is referenced in the package, so that no one can see the actual value. Two calls are made anytime the package is referenced, one to encrypt the data and the other to decrypt.

Transparent Data Encryption encrypts a specific column or an entire tablespace, when the column level is enabled, values of specific columns will be encrypted.

Secure Views. A view is a result set of records, this can be used to secure the data in the database. The view can display subset of a table that the user is interested in, it also reflects the join of tables with multiple data. The view conceals columns and data that should not be seen by specific application. The secure view depends on application contexts, which is a database object type that simply stores name-value pairs in memory. It restricts users from seeing data from another application. But due to its limitation with only selects statements, the system will rather use the more advanced virtual private database which supports all the data manipulation languages.

Table-Level Security. The Administrator will specify which table can be accessed by the student, the administrator shows who has the right to edit, view privileges to a table. The code must be validated against the required and specified table with the appropriate actions. This will avoid wrong queries into wrong tables, the user will in no way have the right to inject any SQL code in the program (SQL injection), this will be achieved in the user groups specifications. The page control setup does not allow any modification unless the user is an administrator and has those privileges.

## Some Components to Setup to Achieve Security

### Router

Securing the router should be much concern to the host, the router can be secured through Password encryption: where users who know the password can have access to the network. MAC filtering : create a filter list with the MAC addresses of all the device that is allowed on the network, if any devices try to access the network and it is not on the list, it is blocked. There are so many other ways to secure a router.

### Firewall

The firewall will control the flow of data between different networks, and manage the network connections. The firewall will be configured so that the user can access the required TCP/IP ports. Assuming the HTTP listener is operating on port 8080, the configured firewall allows HTTP requests on port 8080 only

### Apex Listener

The Apex Listener is the main component to accept the flow of information from the user's browser to the database, it acts like an 'ear', securing it should be top most priority. The Apex listener has to be configured to accept incoming connections through the specified ports, whether 8080/8585 or any combination. The Listener will be deployed on the Glassfish server to boost the security measures.

## CHAPTER FIVE

### CONCLUSION

#### Introduction

With the evolution of computerized systems and the advancement of technology such as mobile phones and tablets in this digital age, having an electronic system is much preferred. The student can connect to the system from anywhere.

The student does not need to physically obtain the manual student planner, in order to keep track of activities in school. The university will not need to print out student planners anymore, once the online version is introduced in school. The school will have a section where it can send some general information that can be posted and updated by the administration only.

The student will be prompted for due dates of homework/project. Thus it will be difficult for the student to forget deadlines. The student will be able to save and print notes as reminders, and save contacts with their pictures. This contact list can be uploaded in a .csv file which will then eliminate manual data entry. The APEX application leverages full Oracle database capabilities and existing SQL and PL/SQL skills.

## Findings

This student planner application will only run on an Oracle database in the administrative and installation back-end. APEX is a sole proprietary application from Oracle, but it encompasses HTML, CSS, and it is capable of running on any browser.

Statistics taken from the interviews conducted with students showed a majority wanted such an application for most of the reasons mentioned in Chapter One. [e.g., (see Figure 27)].

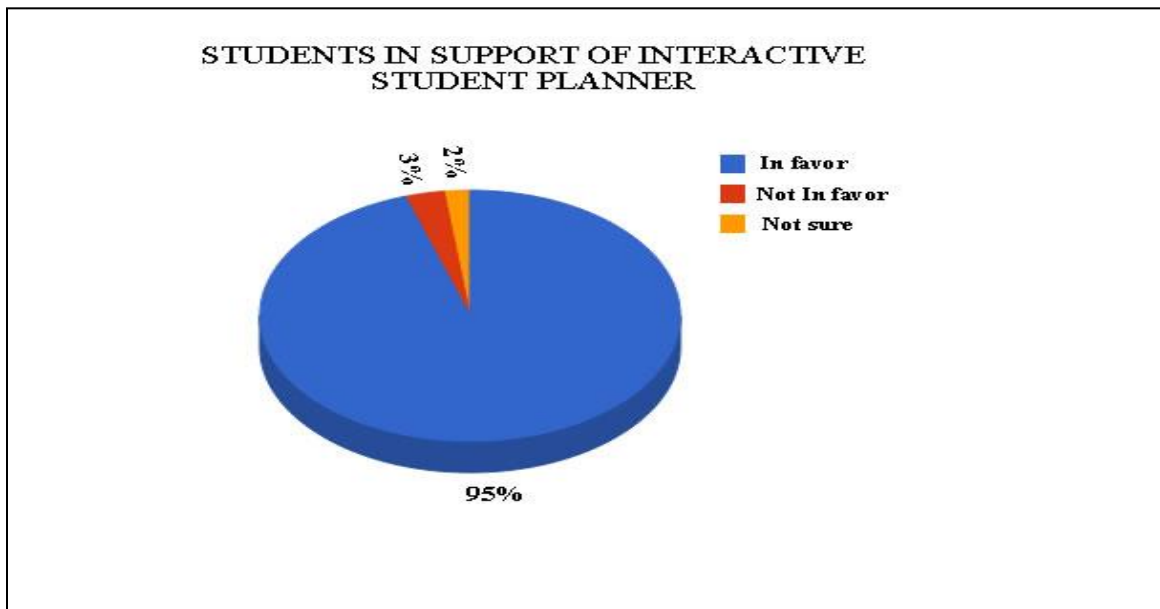


Figure 27. Pie Chart of Students Sample

Most of the difficulties encountered in designing the system were not actually with the requirements of the users. The challenge came from spending much time to program in SQL, PL/SQL and getting the pages to do exactly what was needed. Designing the overall application flow, designing the APEX pages flow, figuring out which component to use where, and how best to implement the business rules or the application logic. Details of the difficulties are discussed below.

Most of the default themes for APEX 4.2 are not so responsive on tmobile phones. This non-responsiveness requires the creation of two separate applications -- one for the desktop and the second for the mobile phone. Themes from bootstrap had to be created and managed in order to be very responsive on both mobile devices and desktop.

Debugging requires knowledge of SQL and PL/SQL, Apex flow and page rendering, HTML and CSS. JavaScript, AJAX, JQUERY were used to enhance the appearance of the application.

Browser differences do not affect the APEX application. However, the calendar and some plug-ins do not work with the earlier theme created in the bootstrap, so the latest version of the bootstrap was used .

The Interactive Report does not work with Oracle 10.1.0.5. But in Oracle 11, the appearance of the form can be changed with CSS. The limitations of the Interactive Report allows only one interactive form to be used on a page. It

allows one set of data manipulation process per page. Although one report is enough to perform all functionalities, a user can make use of an extra Interactive report page for other processing.

The Interactive Report has a row limit which when exceeded will result in an error. To avoid this, either the application will limit the number of report columns or the system will prompt the user for the required size of report columns.



## Lessons

In order for the proposed student planner application to be effective, the school has to run on an Oracle database management system. CSUSB already uses Oracle and PeopleSoft to manage its student data. This is beneficial for this application because only the tables, triggers, sequences, privileges will need to be created and database administration can be left with CSUSB DBA staff.

Any version of Oracle of at least 9i has to be installed in order to use APEX for administrative tasks. The APEX Interactive Report has built-in features -- column selection and ordering, advanced search capability, filters, highlighting, sorting, breaks, aggregates, charting, computations and flashback. These features will enable the user to easily control how data appears.

## Benefits

The Interactive Student Planner application will make handling of information very easy and secured. The student has to be registered in order to use the system. The worst case scenario in losing information is when the database crashes. But the school creates backups, or the developer can use the RMAN capabilities of Oracle to create backup. The highly configurable security features of APEX -- authentication, authorization, and fine-grained security, will make the data secure.

The use of the Interactive Student Planner will help reduce the cost of printing the student planner every quarter, and these expenses can be used to subsidize other projects in the school.

It is easy to sort out the pages in the student planner within seconds as compared to the cumbersome nature of having to flip through the pages of the manual planner.

Finally each student will be allocated space where files can be kept and managed online. This will facilitate easy retrieval and record-keeping. There will be no need to stack up manual planners each year and information will not be lost easily.

## Future Work

Although the application can be used by any student, the developer purposely designed it for Computer Science Students of California State University San Bernardino as the primary clients. In view of that the courses and classes in the lookup table report are the respective ones offered in the schools' department. Any other interested student can edit the course report to suit their needs. Some other functionalities like deploying the application on Android will be considered, the campus map will also be improved to include a search link of the various locations based on the user input. The user will put in a specific location and the application will pin to the spot on the map, this will help with searches since the user will not have to scan through all the list of locations. These will be a future work the developer will look forward to creating.

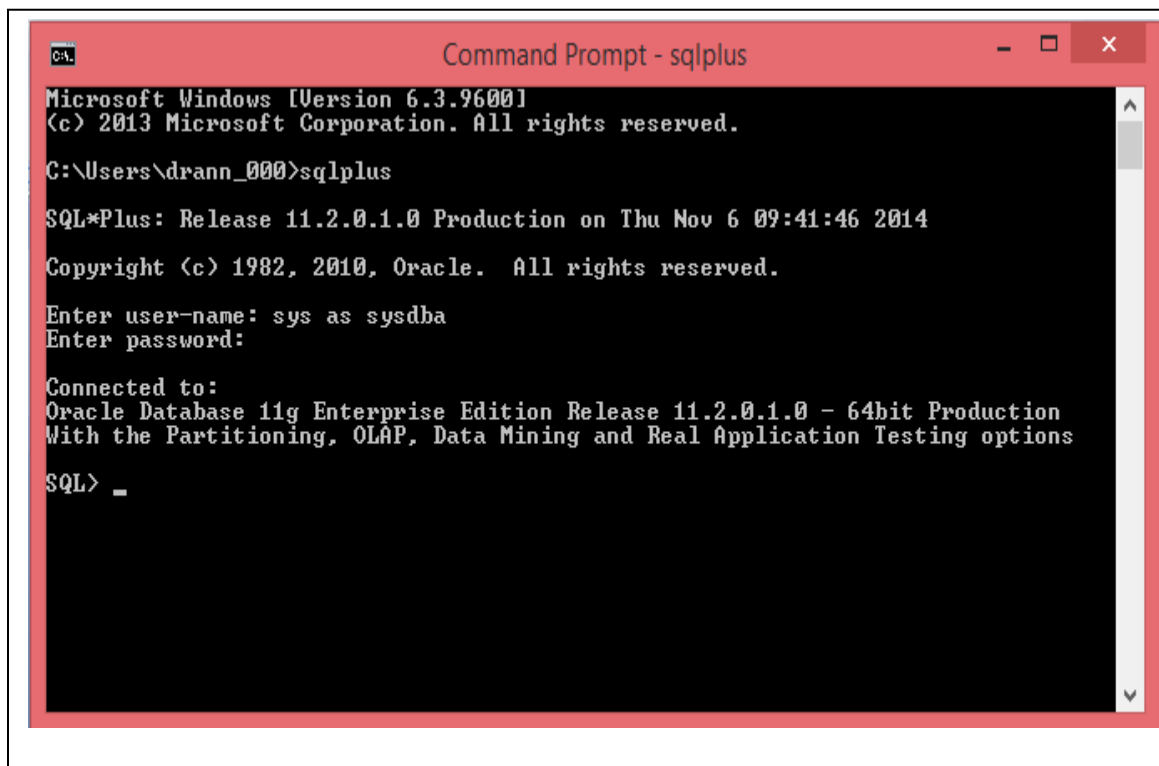
## APPENDIX A

### INSTALLING ORACLE DATABASE 11G ON WINDOWS SERVER 2012

- Download the Express Edition of Oracle 11g Database from <http://www.oracle.com/index.html>, it is two files, Disk 1 and Disk 2, download both. The size is about 2G, select the required Operating system
- Accept the OTN License Agreement, the download will begin
- Unzip both files, name one folder databse1 and the other database2, copy all the stage components from databse2 Disk 2 (database\_2of2) into the stage components in databse1Disk 1(database\_1of2) , this is a very important stage
- Go back into the databse1 folder and double click on the setup, a command prompt shell will pop, preparing to launch Oracle Universal Installer
- The universal Installer opens, there are 9 steps which need to be followed accordingly
- Step 1: Configure Security Updates; Oracle ask for your email address, this helps Oracle to send you security updates, you can skip this step by clicking on the next button, a window will pop up to warn you of the skip, click ok.
- Step 2: Installation Option: Select your installation option, usually do the create and configure a database, click next
- Step 4: Typical Installation: Accept the default
- Step 3: Install Type: There is a option between Desktop Class and Server Class, choose the Desktop Class, it gives minimal configuration, since we are not using Oracle for production purposes,
- Step 5: Prerequisite Checks: You can keep to the default values, or specify your basic configuration, make sure the database edition is set to Enterprise edition which is about 3.34GB, maintain the Global database name, set the administrative password to use to sign into Oracle. A pop window opens up to warn you of the password recommendation standard by Oracle, click Yes and proceed. The prerequisites environment are checked with a progress bar to make sure the minimum configuration can be met.
- Step 6: Summary; A list of the Database information, Global information that was checked during the step 5
- Step 7:Install products; the installation begins, this might take some time, when it is done a Database Configuration Assistant

window pops up, you can do password management to Lock/ unlock database user account or change the default password, ignore that and click ok

- Step 7: Installation is finished with some important note, it gives you the URL for the Enterprise Manager:  
<https://localhost:1158/em>, this link is very important to database Administrator but we can ignore it
- To check that Oracle was successful installed go to command prompt, enter sqlplus as shown in the figure below



```
C:\Users\drann_000>sqlplus

Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\drann_000>sqlplus

SQL*Plus: Release 11.2.0.1.0 Production on Thu Nov 6 09:41:46 2014

Copyright (c) 1982, 2010, Oracle. All rights reserved.

Enter user-name: sys as sysdba
Enter password:

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> _
```

APPENDIX B  
INSTALLING APEX 4.2

- Download APEX 4.2 from the Oracle website  
<http://www.oracle.com/technetwork/developer-tools/apex/downloads/index.html>
- Accept the OTN Licensing Agreement to begin the download, unzip and extract the download file into a required folder usually into the drive C
- Open command prompt, change directory to where the apex file is saved
- Login to the database and create tablespace in the database for APEX
- SQL> CREATE TABLESPACE APPX  
2 DATAFILE 'the path to the database  
name\APPX\_211103.dbf' SIZE 300M  
3 EXTENT MANAGEMENT LOCAL  
4 SEGMENT SPACE MANAGEMENT AUTO;
- You can create a temporary tablespace, although it is not necessary
- SQL> CREATE TEMPORARY TABLESPACE TEMP\_02  
2 TEMPFILE 'the path to the database name\TEMP\_02\_01.dbf'  
SIZE 100M;
- Exit from sqlplus and open a new command prompt, change directory to the root of where the APEX file is saved and log in back to sqlplus from there
- C:\>cd apex  
C:\apex>sqlplus  
Enter the user-name: sys as sysdba  
Enter password:
- Once connected to the database, do this (@apexins.sql tablespace datafile temporal tablespace /the directory for all the images of APEX/ )
- SQL>@apexins.sql APPX APPX TEMP\_02 /i/
- Press enter, installation of APEX will begin, this may take several minutes, make sure you get this successful message
- PL/SQL procedure successfully completed



- Create and load the APEX images into a directory, it should always be one directory up wherever you have your APEX files
- SQL> @apxldimg.sql c:\  
PL/SQL procedure successfully completed  
Directory created
- Now set up the port for the APEX application, Run the configuration script, this would allow you to set the administrator password and set which port the APEX application should listen from
- SQL>@apxconf  

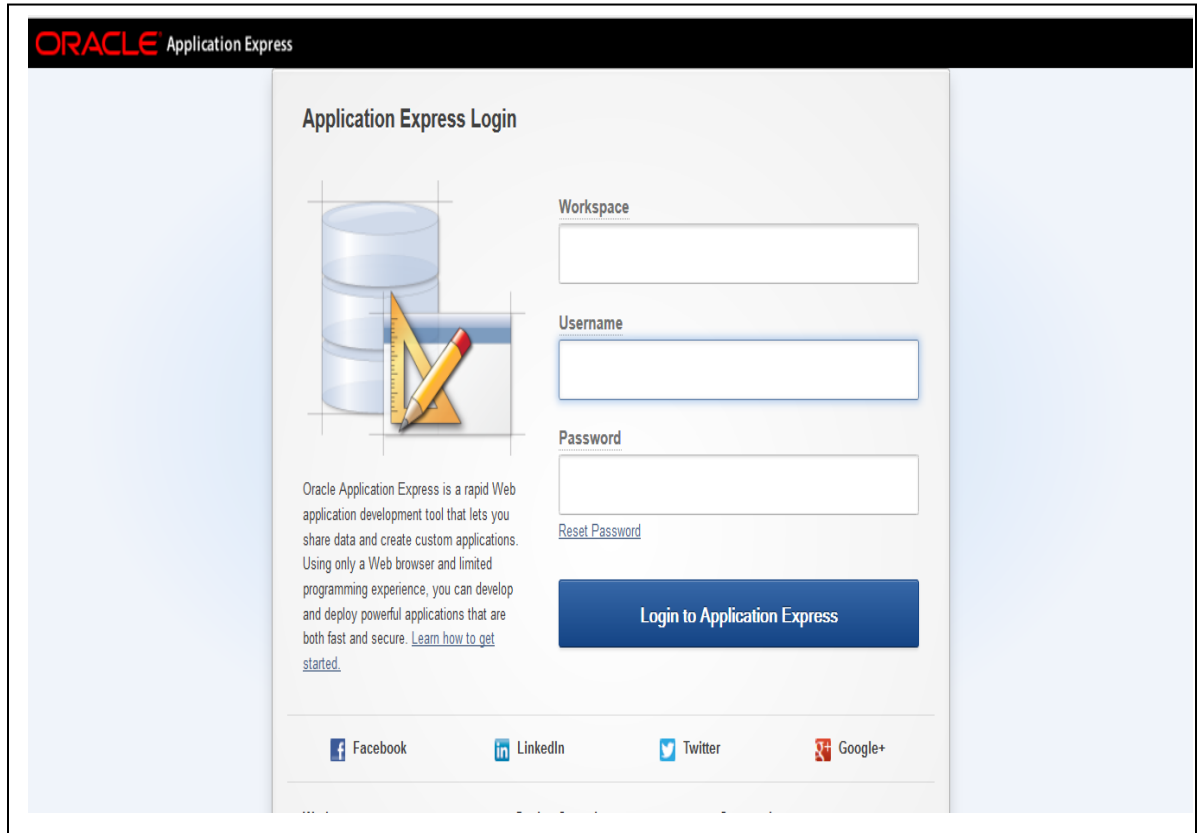
| PORT |
|------|
| 8080 |
- You can accept the default port or change it
- SQL> SELECT DBMS\_XDB.GETHTTPPORT FROM DUAL;  

| GETHTTPPORT |
|-------------|
| 8080        |

SQL> EXEC DBMS\_XDB.SETHTTPPORT(8585);  
PL/SQL procedure successfully completed.  
SQL> SELECT DBMS\_XDB.GETHTTPPORT FROM DUAL;  

| GETHTTPPORT |
|-------------|
| 8585        |
- Set the password and configure it, you can always reset the password from sqlplus by running this command
- SQL> @apxchwd.sql
- Go into any web browser and type , localhost: the port(8080)/apex/apex\_admin
- Once in, follow the instructions, APEX will allow you to change your password
- Create a workspace where all your applications will be saved.
- Logout and Login into APEX, enter the url, localhost: the port(80808)/apex/

- You will see a screen like this



The screenshot shows the Oracle Application Express Login page. At the top, there is a black header with the Oracle logo and the text "Application Express". Below the header, the page is divided into three main sections. On the left, there is a large blue vertical bar. In the center, there is a light gray box containing the "Application Express Login" title, a graphic of a database cylinder and a pencil, and a paragraph of text describing the tool. On the right, there is a white box containing the login form. The form has three input fields: "Workspace", "Username", and "Password". Below the "Password" field is a link for "Reset Password". At the bottom of the form is a blue button labeled "Login to Application Express". At the bottom of the page, there is a horizontal bar with social media links for Facebook, LinkedIn, Twitter, and Google+.

ORACLE Application Express

### Application Express Login

Oracle Application Express is a rapid Web application development tool that lets you share data and create custom applications. Using only a Web browser and limited programming experience, you can develop and deploy powerful applications that are both fast and secure. [Learn how to get started.](#)

Workspace

Username

Password

[Reset Password](#)

Login to Application Express

Facebook LinkedIn Twitter Google+

- Enter the workspace name and password, APEX environment is setup and you are good to go with APEX

## APPENDIX C

### INSTALLING GLASSFISH SERVER

- Download the Glassfish server from the Glassfish website,  
<https://glassfish.java.net/>
- To begin the installation for the Glassfish, you need to install java on your computer, so download the jdk-8u20-windows-x64 from Oracle website,  
<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>
- Click and install the jdk, this will install java
- After the installation, open the command prompt and copy the path location of the glassfish.exe and add the following parameters
- `C:\folder> E:\glassfish-4.0-windows.exe -j "C:\Program Files\Java\jdk1.7.0_051"`
- Press enter, the java 2 runtime environment is launched, the installation will begin by initializing the wizards
- Choose the default settings and keep clicking next, the process are usually Introduction, Installation Type, Installation, Install Directory, JDK Selection, Update Tool, Ready To Install, Progress, Configuration, Domain Info, Config Results, Summary
- Specify the port you wish to use, whether 80, 8080, 8181, 8585. Enter a password for the admin, this can be changed anytime you logged in to glassfish, check the create operating system service for the domain so you don't have to start the glassfish server manually anytime you restart you computer

Introduction  
Installation Type  
Installation  
Install Directory  
JDK Selection  
Update Tool  
Ready To Install  
Progress  
Configuration  
**Domain Info**  
Config Results  
Summary

ORACLE  
Created by Oracle with

Domain Info

Configure the administration settings for the server. Please provide the username and password for the server. You may leave the passwords empty if you would like to configure the server for unauthenticated logins.

Domain Name

Admin Port

Http Port

Username

Password

Reenter Password

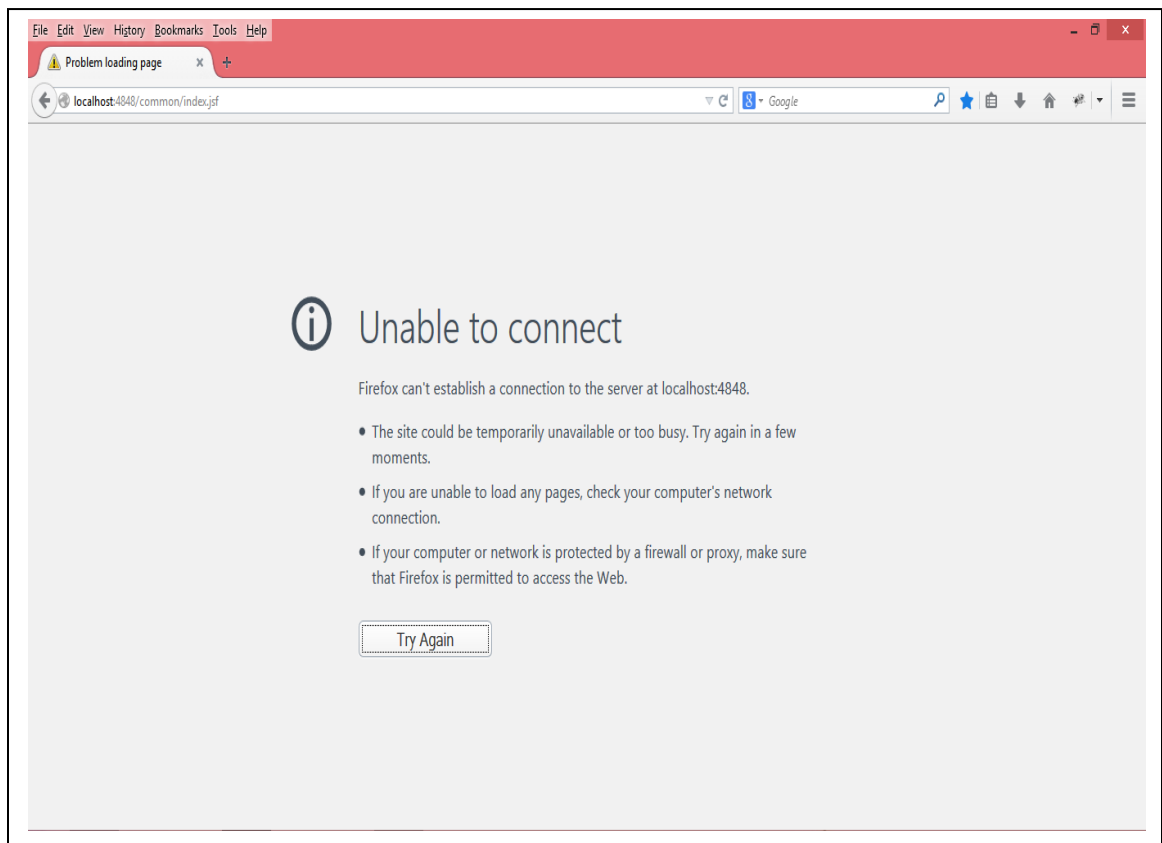
☒ Create Operating System service for the domain

Service Name

☒ Start domain after creation

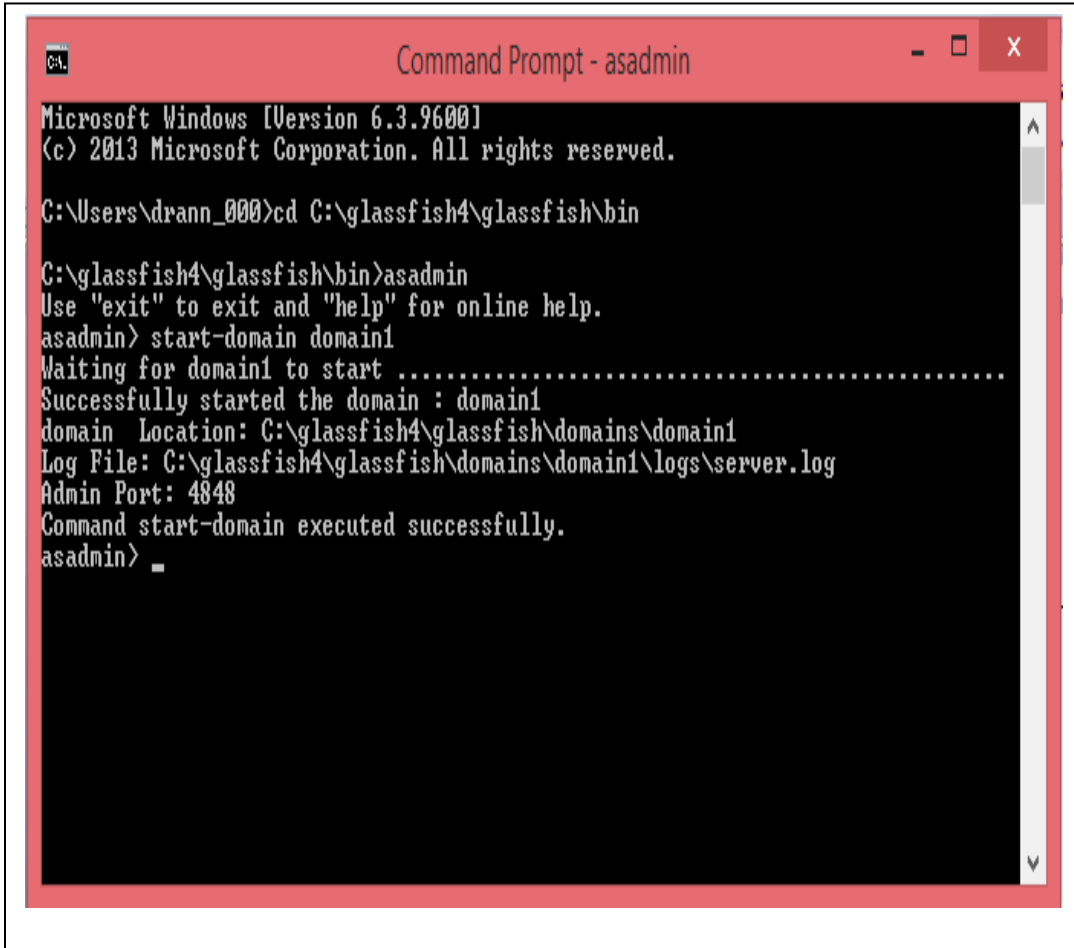
Select this option to create a Operating Sys

- There will be pops up about the Java through the firewall, always allow access
- After the Summary, you can now exit the installation
- Go to services on windows, locate domain1 Glassfish, this will make glassfish start automatically, but if you do not see it in the services, you can always start it manually
- Some common errors that show up when you restart you computer and try to go to the Glassfish console from the browser



This is usually because the glassfish server is not started automatically in the services, or it was not checked during the installation to start automatically.

To resolve this issue, just start the glassfish console manually by changing the directory to the location of the asadmin "C:\glassfish4\glassfish\bin" and start it

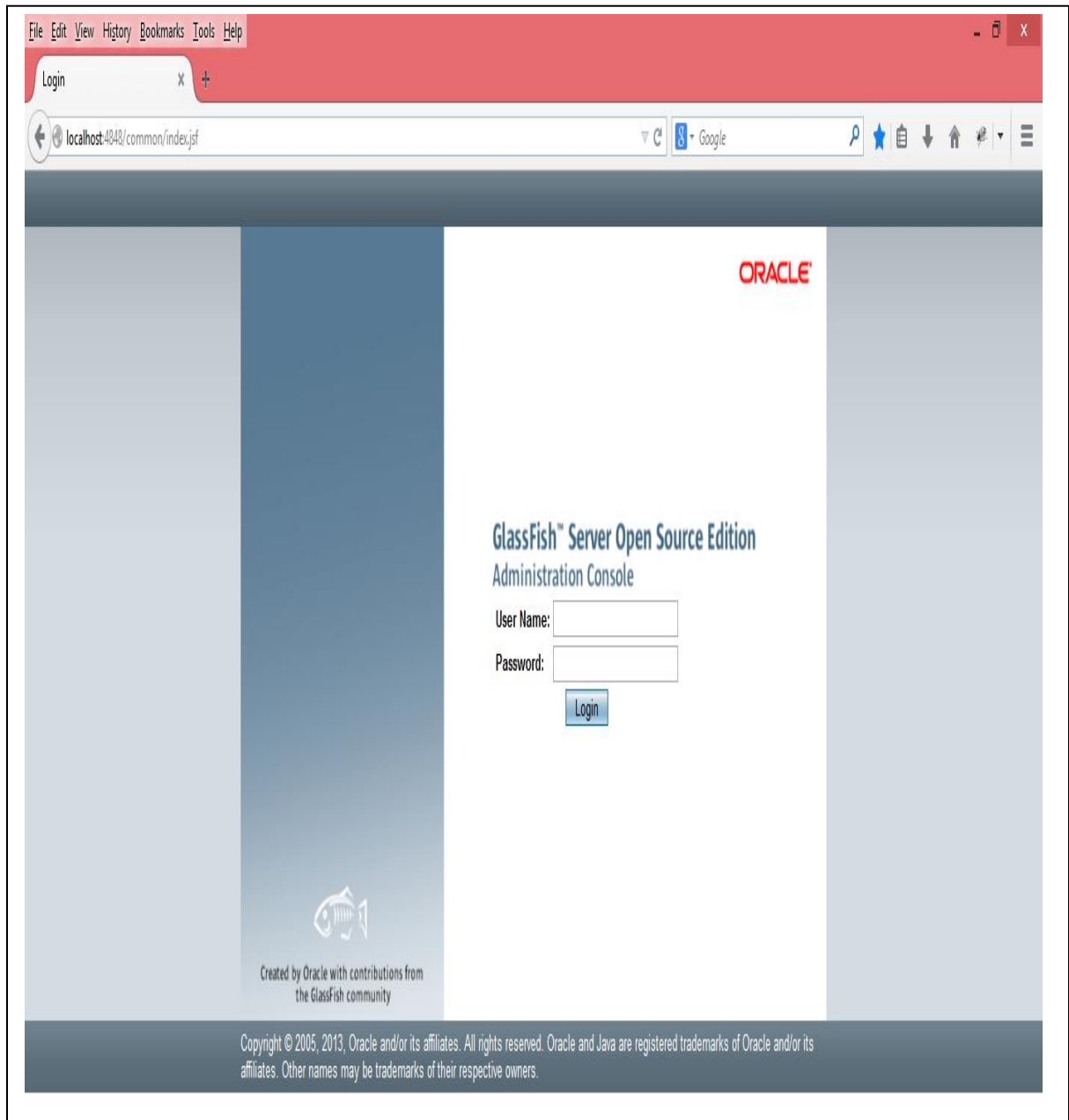


```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

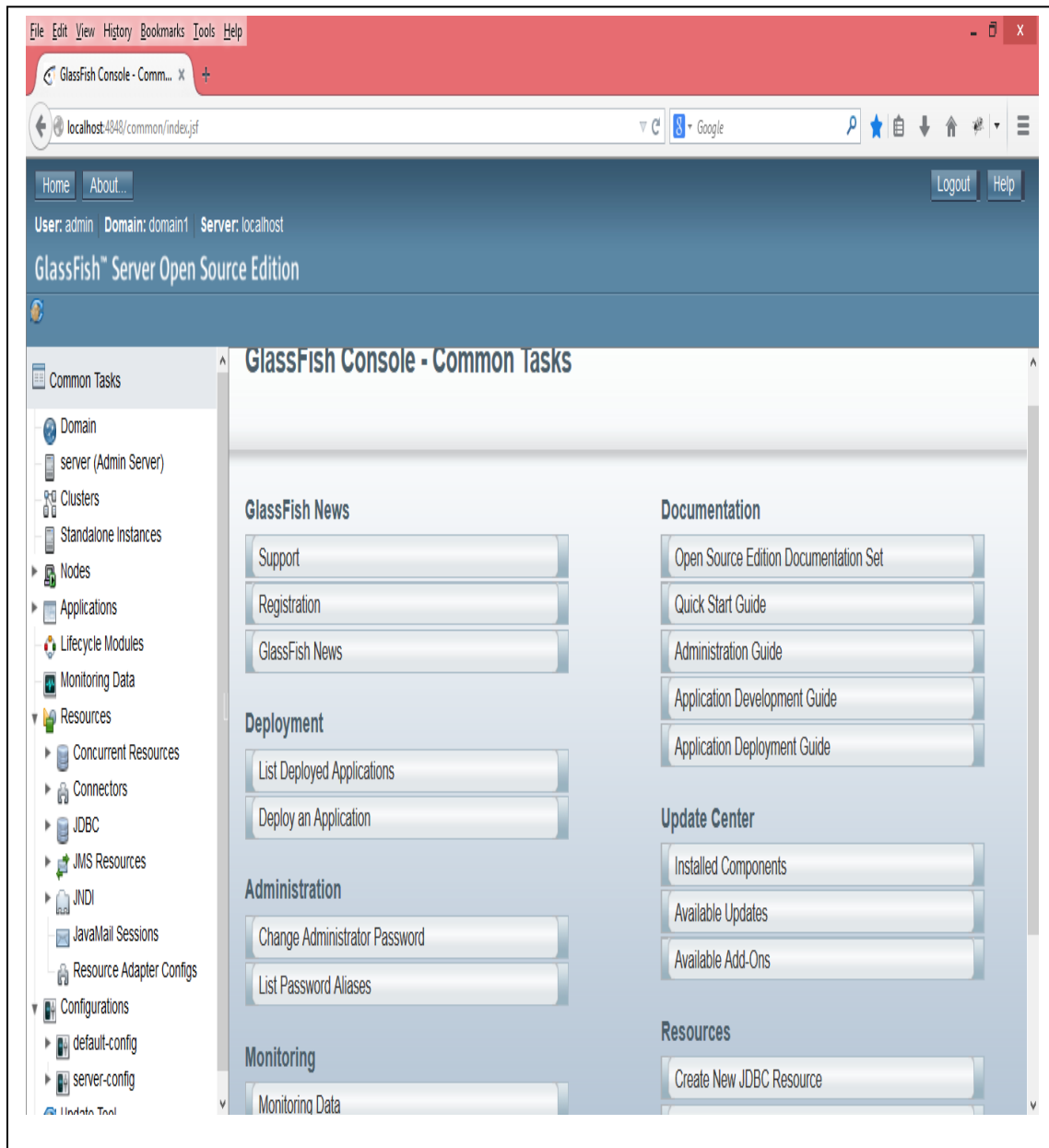
C:\Users\drann_000>cd C:\glassfish4\glassfish\bin

C:\glassfish4\glassfish\bin>asadmin
Use "exit" to exit and "help" for online help.
asadmin> start-domain domain1
Waiting for domain1 to start .....
Successfully started the domain : domain1
domain Location: C:\glassfish4\glassfish\domains\domain1
Log File: C:\glassfish4\glassfish\domains\domain1\logs\server.log
Admin Port: 4848
Command start-domain executed successfully.
asadmin> _
```

Go to your browser of choice and enter "http://localhost:4848/", If the Glassfish is started, you will see the Login screen



Logon using the admin credentials and it takes you to this screen





- Now that the Glassfish is up, you need to deploy the apex listener but you will use the Oracle Rest Data Services (ords), which has more capabilities, download it from <http://www.oracle.com/technetwork/developer-tools/rest-data-services/overview/index.html>

Change the ords.war in the ords folder into apex.war

Set up the war file from command prompt

```
C:\ords>java -jar apex.war setup
```

```
Oct 03, 2014 3:50:00 PM
```

```
oracle.dbtools.common.config.file.ConfigurationFolder lo
```

```
gConfigFolder
```

```
INFO: Using configuration folder: C:\ords\config\apex
```

```
Enter the name of the database server [localhost]:
```

```
Enter the database listen port [1521]:
```

```
Enter 1 to specify the database service name, or 2 to specify the  
database SID [
```

```
1]:1
```

```
Enter the database service name:orcl
```

```
Enter the database user name [APEX_PUBLIC_USER]:
```

```
Enter the database password for APEX_PUBLIC_USER:
```

```
Confirm password:
```

```
Enter 1 to enter passwords for the RESTful Services database users  
(APEX_LISTENER,APEX_REST_PUBLIC_USER), 2 to use the same password as  
used for APEX_PUBLIC_USE
```

```
R or, 3 to skip this step [1]:2
```

```
Oct 03, 2014 3:52:55 PM
```

```
oracle.dbtools.common.config.file.ConfigurationFiles upd
```

```
ate
```

```
INFO: Updated configurations: defaults, apex, apex_al, apex_rt
```

```
C:\ords>java -jar apex.war static
```

```
C:\glassfish4\glassfish\domains\domain1\docroo
```

t\i

WAR Generation complete

WAR location : C:\ords\i.war

Context path : /i

Static resources : C:\glassfish4\glassfish\domains\domain1\docroot\i

Ensure the static resources are available at path:

C:\glassfish4\glassfish\domai

ns\domain1\docroot\i

on the server where the WAR is deployed

- Connect to sqlplus and execute this
- SQL> select dbms\_xdb.gethttpport() from dual;  
DBMS\_XDB.GETHTTPPORT()  
-----  
8080
- SQL> exec dbms\_xdb.sethttpport('0');  
PL/SQL procedure successfully completed.
- SQL> select dbms\_xdb.gethttpport() from dual;  
DBMS\_XDB.GETHTTPPORT()  
-----  
0
- SQL> alter user APEX\_PUBLIC\_USER identified by 123456;  
User altered.
- Copy all the images from the apex folder images into  
C:\glassfish4\glassfish\domains\domain1\docroot\i, the i folder  
was created during the war generation
- Configure the glassfish server, navigate to the  
Configurations/server-config/Security/Realms/File, choose Manage  
users and add the users



File Edit View History Bookmarks Tools Help

File UsersOracle REST Data Services

localhost:4848/common/index.jsf

Google

HomeAbout...LogoutHelp

User: adminDomain: domain1Server: localhost

GlassFish™ Server Open Source Edition

Total # of available updates : 48

Server Settings

Admin ServiceConnector ServiceEJB ContainerHTTP ServiceJVM SettingsJava Message ServiceLogger SettingsMonitoringNetwork ConfigORBSecurityRealmsadmin-realmcertificatefileAudit ModulesJACC ProvidersMessage SecuritySystem PropertiesThread Pools

File Users

Manage user accounts for the currently selected security realm.

Configuration Name: server-config

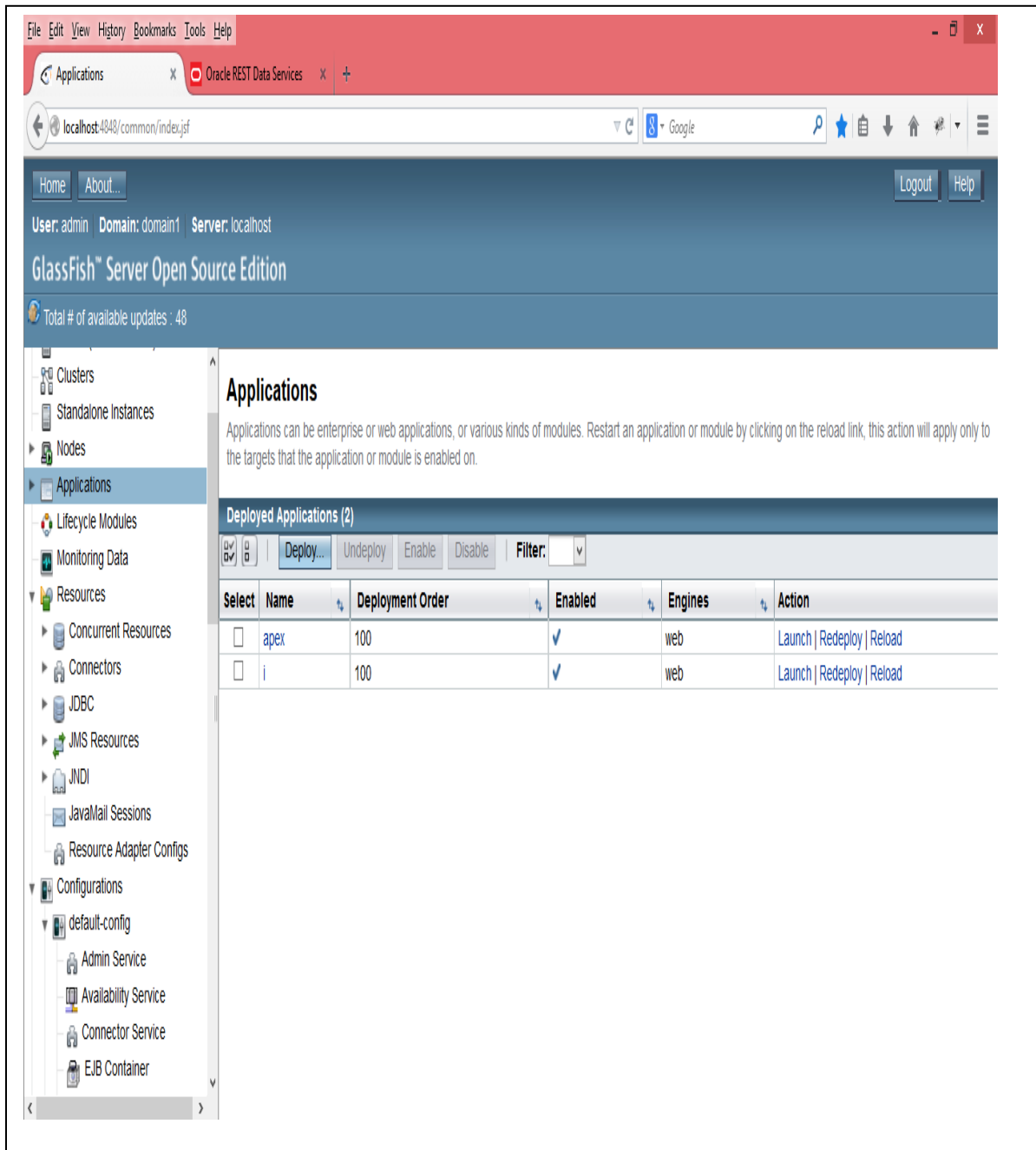
Realm Name: file

File Users (2)

New...Delete

| Select                   | User ID         | Group List |
|--------------------------|-----------------|------------|
| <input type="checkbox"/> | managerlistener | Manager    |
| <input type="checkbox"/> | adminlistener   | Admin      |

- Deploy the APEX Listener as an application in the Glassfish Server, navigate to the Applications node and select the Deploy button, go to the ords folder and select the apex.war. Select the i folder and deploy as well, in the example below the apex, and i file has already been deployed.



The screenshot shows the GlassFish Server Open Source Edition web console. The browser address bar indicates the URL is `localhost:4848/common/index.jsf`. The page title is "GlassFish™ Server Open Source Edition". The left sidebar shows the navigation tree with "Applications" selected. The main content area is titled "Applications" and contains a table of deployed applications.

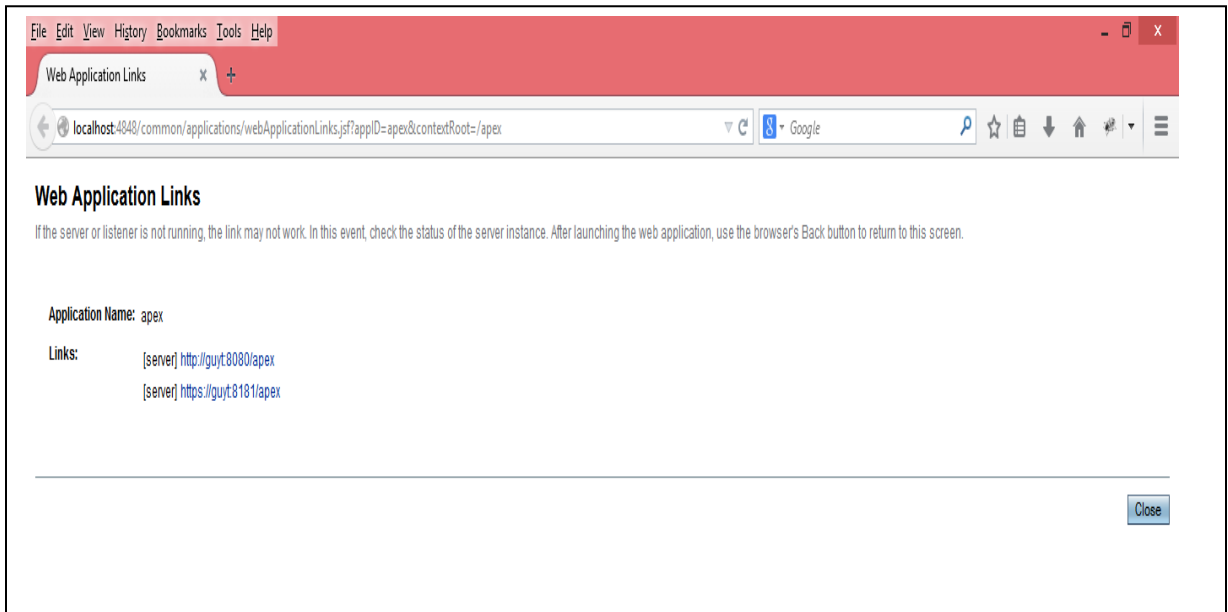
**Applications**

Applications can be enterprise or web applications, or various kinds of modules. Restart an application or module by clicking on the reload link, this action will apply only to the targets that the application or module is enabled on.

**Deployed Applications (2)**

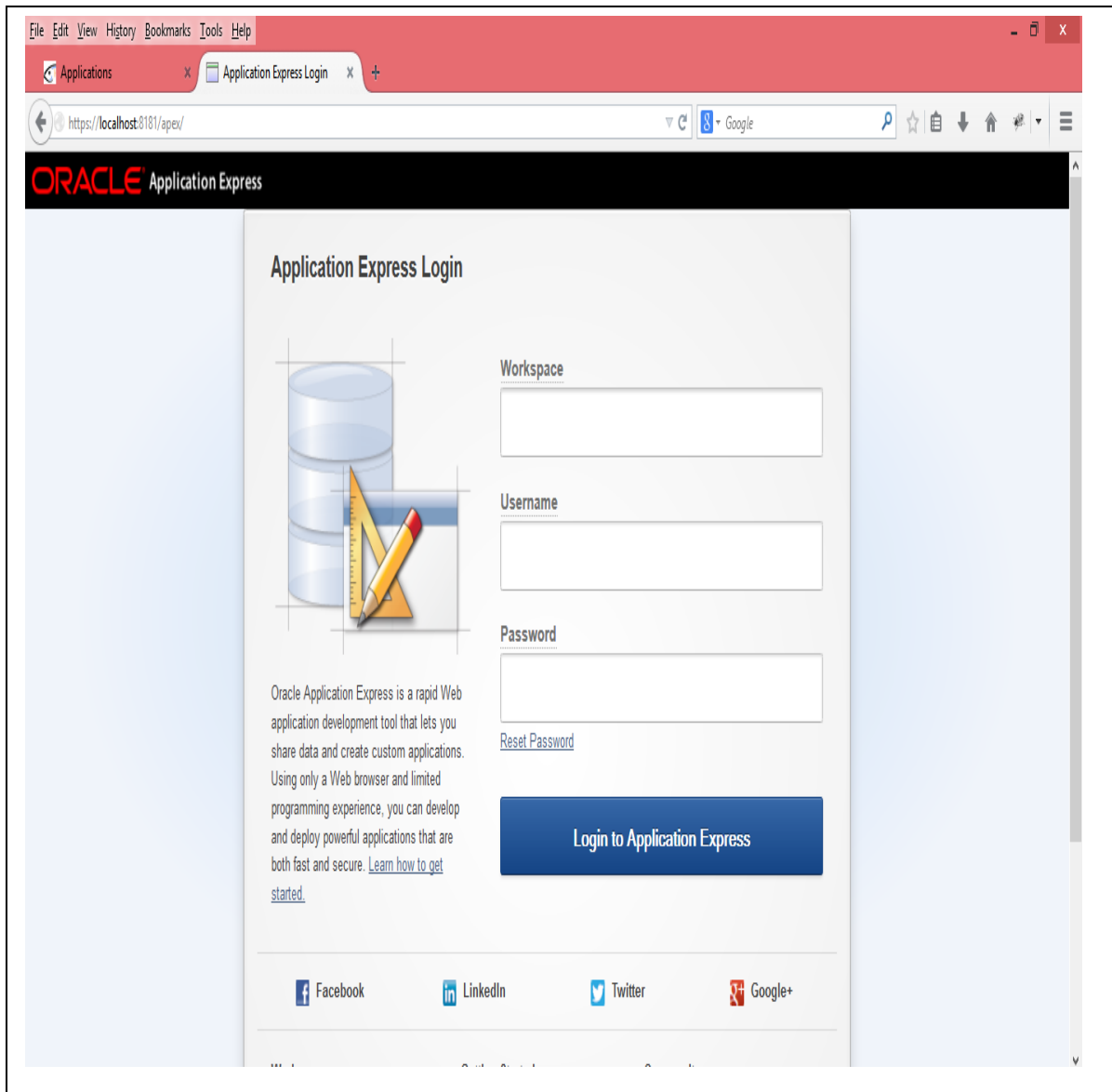
| Select                   | Name | Deployment Order | Enabled | Engines | Action   |
|--------------------------|------|------------------|---------|---------|--|
| <input type="checkbox"/> | apex | 100              | ✓       | web     | <a href="#">Launch</a>   <a href="#">Redeploy</a>   <a href="#">Reload</a> |
| <input type="checkbox"/> | i    | 100              | ✓       | web     | <a href="#">Launch</a>   <a href="#">Redeploy</a>   <a href="#">Reload</a> |

- Click on the launch it will bring you to this screen, click on the desired url, but in our case, click on the https for a more secured connection



https://guyt:8181/apex, this is in the form of  
https://localhost:8181/apex, where localhost is the machine on which  
the APEX is installed

- Once you click on the url, it will send you to the APEX logon screen, this means that APEX is running on the Glassfish server now. You cannot run apex without starting the Glassfish server.



## APPENDIX D

### DATABASE OBJECT SUMMARY AND HOW TO CREATE THE SCHEMA



#### DATABASE OBJECT SUMMARY

| OBJECT TYPE | COUNT |
|-------------|-------|
| INDEX       | 16    |
| PACKAGE     | 4     |
| PROCEDURE   | 2     |
| FUNCTIONS   | 2     |
| SEQUENCE    | 7     |
| TABLE       | 12    |
| TRIGGER     | 7     |

4 main database packages have to be installed in order to use the bootstrap theme, the components can be downloaded from the bootstrap website and customized to suit the application. The packages are S4ATB\_FLOT\_PCK, S4ATB\_LIST\_PCK, S4ATB\_MORRIS\_PCK, anyone interested in using the theme for development can send me an email for help on how to configure the packages

The schema name is planner which will hold the tables for the student planner application.

Connect to the database, create user and grant all privileges to the user.

Connected to:  
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit  
Production  
With the Partitioning, OLAP, Data Mining and Real Application Testing  
options

```
SQL> sho user
USER is "SYSTEM"
SQL> select * from system_privilege_map
      2  where name like '%PRIV%';
```

| PRIVILEGE NAME                  | PROPERTY |
|---------------------------------|----------|
| -167 GRANT ANY PRIVILEGE        | 0        |
| -244 GRANT ANY OBJECT PRIVILEGE | 0        |

```
SQL> -- Create a new user with just and grant
SQL> -- any privilege to, well grant all privileges.
SQL> create user planner identified by planner;
```

User created.

```
SQL> grant create session, grant any privilege to planner;
```

Grant succeeded.

```
SQL> grant all privileges to planner;
```

Grant succeeded.

```
SQL> grant dba to planner;
```

Grant succeeded.

APPENDIX E

HOW TO CREATE THE TRIGGERS

Trigger populate the primary key value when a row is inserted; a procedural code that is automatically executed in response to certain events on a particular table in a database. INSERT event, UPDATE event, DELETE event

This script creates triggers for populating the primary key value in each table. The primary key is based on the sequence created for that field in the table.

### **SQL SCRIPTS**

```
/* Trigger for inserting a sequence value in the PROJECTS table. */
CREATE OR REPLACE TRIGGER "PLANNER"."BI_PROJECT"
    before insert on PROJECT
    for each row
begin
    if :NEW.PROJECT_ID is null then
        select PROJECT_ID_SEQ.nextval into :NEW.PROJECT_ID from dual;
    end if;
end;
/
ALTER TRIGGER "PLANNER"."BI_PROJECT" ENABLE;
/* Trigger for inserting a sequence value in the COURSES table. */
CREATE OR REPLACE trigger BI_COURSES
    before insert on COURSES
    for each row
begin
    if :NEW.COURSE_ID is null then
        select COURSE_ID_SEQ.nextval into :NEW.COURSE_ID from dual;
    end if;
```

```

end;

/

/* Trigger for inserting a sequence value in the homework table. */

CREATE OR REPLACE TRIGGER "PLANNER"."BI_HOMEWORK"

before insert on HOMEWORK

for each row

begin

    if :NEW.HOMEWORK_ID is null then

        select HOMEWORK_ID_SEQ.nextval into :NEW.HOMEWORK_ID from dual;

    end if;

    if inserting or updating then

        :NEW.CREATED      := localtimestamp;

        :NEW.CREATED_BY := nvl(v('APP_USER'),USER);

        :NEW.UPDATED      := localtimestamp;

        :NEW.UPDATED_BY := nvl(v('APP_USER'),USER);

    end if ;

end;

/

ALTER TRIGGER "PLANNER"."BI_HOMEWORK" ENABLE;


/* Trigger for inserting a sequence value in the REPORT table. */

CREATE OR REPLACE trigger BI_REPORT

before insert on REPORT

for each row

begin

    if :NEW.REPORT_ID is null then

        select REPORT_ID_SEQ.nextval into :NEW.REPORT_ID from dual;

    end if;

end;

end;

```

```

/

/* Trigger for inserting a sequence value in the HW_SCORES table. */
CREATE OR REPLACE trigger BI_HW_SCORES
    before insert on HW_SCORES
    for each row
begin
    if :NEW.HW_SCORE_ID is null then
        select HW_SCORE_ID_SEQ.nextval into :NEW.HW_SCORE_ID from dual;
    end if;
end;
/

/* Trigger for inserting a sequence value in the CONTACTS table. */
CREATE OR REPLACE TRIGGER "PLANNER"."BI_CONTACT"
    before insert on CONTACT
    for each row
begin
    if :NEW.CONTACT_ID is null then
        select CONTACT_ID_SEQ.nextval into :NEW.CONTACT_ID from dual;
    end if;
end;
/

ALTER TRIGGER "PLANNER"."BI_CONTACT" ENABLE;
CREATE OR REPLACE TRIGGER "PLANNER"."BI_NOTE"
    before insert on NOTE
    for each row
begin
    if :NEW.NOTE_ID is null then

```

```

        select NOTE_ID_SEQ.nextval into :NEW.NOTE_ID from dual;
    end if;
end;
/
ALTER TRIGGER "PLANNER"."BI_NOTE" ENABLE;
CREATE OR REPLACE TRIGGER "PLANNER"."BIU_PROJECT_FILES"
BEFORE INSERT OR UPDATE ON PROJECT_FILES
for each row
BEGIN
    if :new.ID is null then
        select to_number(sys_guid(),'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX')
into :new.id from dual;
    end if;
    if inserting then
        :new.created := localtimestamp;
        :new.created_by := nvl(wwv_flow.g_user,user);
        :new.updated := localtimestamp;
        :new.updated_by := nvl(wwv_flow.g_user,user);
        :new.row_version_number := 1;
    elsif updating and :new.filename is not null and
nvl(dbms_lob.getlength(:new.file_blob),0) > 15728640 then
        raise_application_error(-20000, 'The size of the uploaded file
was over 15MB. Please upload a smaller sized file.');
```

```

    elsif updating then
        :new.row_version_number := nvl(:old.row_version_number,1) + 1;
    end if;
    if inserting or updating then
        :new.updated := localtimestamp;
        :new.updated_by := nvl(wwv_flow.g_user,user);
    end if;

```

```
END;
```

```
/
```

```
ALTER TRIGGER "PLANNER"."BIU_PROJECT_FILES" ENABLE;
```



APPENDIX F

HOW TO CREATE THE FUNCTIONS

The function created will be used for the Authentication of a user, I created a table to hold the list of users. The password field is encrypted by a hash function

```
create or replace function cauth (p_username in VARCHAR2, p_password in VARCHAR2)
```

```
return BOOLEAN
```

```
is
```

```
    l_password varchar2(4000);
```

```
    l_stored_password varchar2(4000);
```

```
    l_count number;
```

```
begin
```

```
-- First, check to see if the user is in the employees table
```

```
select count(*) into l_count from my_users u where upper(user_name) = upper(p_username);
```

```
if l_count > 0 then
```

```
    -- First, we fetch the stored hashed password
```

```
select user_pwd into l_stored_password
```

```
from my_users where upper(user_name) = upper(p_username);
```

```
    -- we have to apply the custom hash function to the password
```

```
    --l_password := chash(p_username, p_password);
```

```
    l_password := p_password;
```

```
    -- Finally, we compare them to see if they are the same and return
```

```
    -- either TRUE or FALSE
```

```
    if l_password = l_stored_password then
```

```
        return true;
```

```
    else
```

```
        return false;
```

```
    end if;
```

```

end if;

end cauth;

/*Applying the hashing function to the password*/

create or replace function chash (p_username in varchar2, p_password in
varchar2)

return varchar2

is

    l_password varchar2(4000);

    l_salt varchar2(4000) := 'MS5XWMK9XG6NFDS454FFNFD3LJK08Q';

begin

-- This function should be wrapped, as the hash algorithm is exposed
here.

-- You can change the value of l_salt or the method of which to call
the

-- DBMS_OBFUSCATION toolkit, but you must reset all of your passwords
-- if you choose to do this.


l_password := utl_raw.cast_to_raw(dbms_obfuscation_toolkit.md5
    (input_string => p_password || substr(l_salt,10,13) || p_username ||
        substr(l_salt, 4,10)));

return l_password;

end;

```

## APPENDIX G

CODE FOR SOME SECTION OF THE HOMEPAGE

Homework report is a PL/SQL anonymous block with the following region source

```
declare

    c pls_integer := 0;

    d date;

    l_ts timestamp with local time zone;

    l_pastdue_displayed boolean := false;

    l_upcoming_displayed boolean := false;

    l_days_diff number;

    l_color varchar2(30);

    max_rows integer := 10;

    l_text    varchar2(4000) := null;

begin

    d := sysdate;

    l_ts := localtimestamp;


    sys.htp.p('<div class="panel-body">');

    for c1 in (

select (case

    when length(HOMEWORK_DESCRIP) > 50 then

        substr(HOMEWORK_DESCRIP,1,30) || '...' ||

        substr(HOMEWORK_DESCRIP,length(HOMEWORK_DESCRIP) - 20)

    else HOMEWORK_DESCRIP

end) description,

    due_date,

    homework_id

FROM homework

    where userid =( select userid from users where username = :APP_USER)
```

```

order by due_date) loop

    c := c + 1;

    l_days_diff :=
round(to_date(to_char(nvl(c1.due_date,c1.due_date),'DD-MON-YYYY'),'DD-
MON-YYYY') - sysdate);

    --

    l_text := null;

    if l_days_diff < 0 then

        l_text := '<span class="badge badge-
important">'||abs(l_days_diff)||' '||'days overdue'||'</span>';

        --l_color := 'Red';

    elsif l_days_diff = 0 then

        l_text := '<span class="badge badge-
warning">'||'Today'||'</span>';

        --l_color := 'Yellow';

    elsif l_days_diff = 1 then

        l_text := '<span class="badge badge-
success">'||abs(l_days_diff)||' '||'day left'||'</span>';

        --l_color := 'Green';

    else

        l_text := '<span class="badge badge-
success">'||abs(l_days_diff)||' '||'days left'||'</span>';

        -- l_color := 'Green';

    end if;

    --

    /* sys.http.p('<div class="list-group">');

        sys.http.p('<a class="list-group-item" href="#">'||

            l_text|| c1.description || ' </a>');

    */

    sys.http.p('<div class="list-group">');

```

```

        sys.http.p('<a class="list-group-item"
href="'||apex_util.prepare_url('f?p='||:app_id||':13:'

||:app_session||':NO:13,RP:P13_homework_ID:'||c1.homework_id)||'">'||

        l_text|| c1.description ||

        '</a>');

--

if c = max_rows then
    exit;
end if;
end loop;

sys.http.p('</div>');

if c = 0 then
    http.p('No requests found');
end if;

end;

S4ATB Morris Donut Chart [Plug-in],region source
select COURSE_ID as "label",
SUM(SCORES) as "value"
from hw_scores
group by COURSE_ID

```

## APPENDIX H

CODE FOR SOME SECTION OF THE COURSE PAGE



Create an interactive report with a region source

```
select "COURSE_ID",  
APEX_ITEM.CHECKBOX(1,COURSE_ID) ID,  
"COURSE_NAME",  
"QUARTER",  
"NUMBER_OF_UNITS",  
"TYPE"  
from "#OWNER#". "COURSES"
```

The cart is an SQL query of region source

```
SELECT c001, count(c001) as count  
FROM apex_collections  
WHERE collection_name = 'ORDER1'  
group by c001
```

The student courses is SQL Query with region source

```
select "COURSE_NBR",  
"COURSE_NAME",  
"QUARTER",  
"NUMBER_OF_UNITS",  
"TYPE"  
from "#OWNER#". "STUDENT_COURSES"
```

Create the required buttons, make sure that the grid length is set appropriately, customize the user interface

## APPENDIX I

CODE FOR SOME SECTION OF THE PROJECT SUMMARY PAGE

PL/SQL anonymous block for the project summary, code for the region source

```
declare

    c                pls_integer := 0;

    d                timestamp with local time zone;

    l_pastdue_displayed  boolean := false;

    l_upcoming_displayed boolean := false;

    l_days_diff        number;

    max_rows integer := 20;

    l_status           varchar2(30);

begin

    d := localtimestamp;

    sys.http.p('<div class="panel-body">');

    for c1 in (

        select

            c.project_id,

            --nvl(updated,created) updated_on,

            c.due_date,

            c.description,

            c.updated,

            c.project_name,

            c.category

        from   project c

        where  userid =( select userid from users where username =

:APP_USER)

        order by  c.due_date desc
```

```

    )

    loop

    c := c + 1;

    sys.http.prn('</ul><div class="clear"></div></div>');

    sys.http.p('<div class="timelineStatusList">');

    sys.http.p(' <a class="list-group-item"
href="'||apex_util.prepare_url('f?p='||:app_id||':16:'||:app_session||'
::NO:16,RP:P16_PROJECT_ID:'||c1.project_id)||'">');

    sys.http.p('');

/*

    sys.http.prn('      <h4><a title="View meeting -
'||apex_escape.html(c.short_meeting_name)||'"
href="'||apex_util.prepare_url('f?p='||:APP_ID||':70:'||:APP_SESSION||'
:::70:P70_MEETING_ID,P70_SECTION_ID,P70_CAME_FROM:'||apex_escape.html(c
.meeting_id)||',1,'||:APP_PAGE_ID)||'"
>'||apex_escape.html(c.meeting_name)||'</a></h4>'||

    '      <p>'||

*/

/* sys.http.p(' <a class="list-group-item"
href="'||apex_util.prepare_url('f?p='||:app_id||':29:'||:app_session||'
::NO:29,RP:P68_PROJECT_ID:'||c1.project_id)||'">');

    sys.http.p(''); */

    sys.http.p('      <span
style="color:red">'||apex_escape.html(substr(c1.project_name,1,150))||

    '      </span>      <p>'||

    '      <span><strong>Last Updated: </strong>
'||apex_util.get_since(c1.updated)

```

```

||'</span><br>'||

'          <span><strong>Due Date(s): </strong>
'||apex_escape.html(c1.due_date)||'</span><br>'||

'          <span><strong>Category: </strong>
'||apex_escape.html(c1.category)||'</span><br>'||

'          <span><strong>Description: </strong>
'||apex_escape.html(c1.description)||'</span>');

sys.http.prn('          </p>');

sys.http.prn('      </span>');

sys.http.prn('</a>');

if c = max_rows then
    exit;
end if;

end loop;

sys.http.p('</div>');

if c = 0 then
    http.p('No recent feedback found');
end if;

end;

Set all validations, set grid to appropriate layout, position and
customize the buttons as preferred. Set the print button to redirect to
url "javascript:window.print();" and execute validations to "YES"

```

## APPENDIX J

CODE FOR SOME SECTION OF THE PROJECT DETAIL PAGE

PL/SQL anonymous block for the project details page, region source code

```
declare

    l_project_name      varchar2(255);
    l_due_date          varchar2(25);
    l_location           varchar2(100);
    l_conf_phone         varchar2(25);
    l_conf_info          varchar2(25);
    l_description        varchar2(4000);

begin

    -- Get the Meeting Details

        select

            project_name,

            due_date,

            description

        into

            l_project_name,

            l_due_date,

            l_description

        from

            project

        where

            project_id = :P16_PROJECT_ID;

    -- Display the Meeting Details section

    sys.http.p('<section id="details" class="uBorderlessRegion clearfix"
aria-live="polite">');

    sys.http.p('  <div class="uRegionHeading">');
```

```

sys.ftp.p(' </div>');

sys.ftp.p(' <div class="uRegionContent clearfix">');

sys.ftp.p(' <ul class="vapList">');

sys.ftp.p(' <span><strong>

    Project Name :

</strong>

<span>

    '||l_project_name||'

</span>

</span><br>');

sys.ftp.p(' <span><strong>

    Due_Date :

</strong>

<span>

    '||l_due_date||'

</span>

</span><br>');

if l_description is not null then

    sys.ftp.p(' <span> <strong>

        Description :

</strong>

<span>

    '||apex_escape.html(l_description)||'

```



```

        </span>
    </span>');
end if;

sys.http.p('    </ul>');

    sys.http.prn('    </div>');

    sys.http.prn('</section>');

end;

```

For the files download report, choose an interactive form and set the region source

```

select
    ID,
    ROW_VERSION_NUMBER,
    PROJECT_ID,
    FILENAME,
    FILE_MIMETYPE,
    FILE_CHARSET,
    FILE_BLOB,
    FILE_COMMENTS,
    TAGS,
    CREATED,
    CREATED_BY,
    UPDATED,
    UPDATED_BY,
    dbms_lob.getlength(file_blob) file_size,
    dbms_lob.getlength(file_blob) download
from PROJECT_FILES

```

## APPENDIX K

CODE FOR SOME SECTION OF THE CONTACT PAGE

Set the region source of the interactive report

```
select "CONTACT_ID",
"FIRST_NAME",
"LAST_NAME",
"GENDER",
"EMAIL",
"PHONE",
'' "PHOTO",
"FILENAME",
"MIMETYPE",
"IMAGE_LAST_UPDATE"
from "#OWNER#"."CONTACT"

where userid =( select userid from users where username = :APP_USER)
```

## APPENDIX L

CODE FOR SOME SECTION OF THE NOTE SUMMARY PAGE

PL/SQL anonymous block for the region source

```
declare

    c                pls_integer := 0;

    d                timestamp with local time zone;

    l_pastdue_displayed    boolean := false;

    l_upcoming_displayed  boolean := false;

    l_days_diff          number;

    max_rows integer := 20;

    l_status            varchar2(30);

begin

    d := localtime;

    sys.http.p('<div class="panel-body">');

    for c1 in (

        select

            c.note_id,

            -- nvl(updated,created) updated_on,

            c.date_created,

            c.description,

            c.note_name,

            c.updated

        from   note c

            where userid =( select userid from users where username =

:APP_USER)

        -- order by nvl(c.note_name,c.date_created) desc

        ORDER BY  c.date_created DESC

    )
```

```

)

loop

c := c + 1;

sys.http.prn('</ul><div class="clear"></div></div>');

sys.http.p('<div class="timelineStatusList">');

sys.http.p('    <a class="list-group-item"

href="||apex_util.prepare_url('f?p=||:app_id||:30:||:app_session||:
:NO:30,RP:P30_NOTE_ID:||c1.note_id)||">');

    sys.http.p('');

--sys.http.p('                <span
style="color:red">||apex_escape.html(substr(c1.note_name,1,150))||

    sys.http.p('                <span class="label label-
warning">||apex_escape.html(substr(c1.note_name,1,150))||

        '    </span>    <p>||

        -- '                <span><strong>Last Updated: </strong>
'||'</span><br>'||

            '                <span><strong>Last Updated: </strong> |||
apex_util.get_since(c1.updated)

        ||'</span><br>'||

            '                <span><strong>Date Created(s): </strong>
'||apex_escape.html(c1.date_created)||'</span><br>'||

            '                <span><strong>Description: </strong>
'||apex_escape.html(c1.description)||'</span>');

sys.http.prn('    </p>');

```

```

sys.http.prn('    </span>');

--<span class="badge badge-warning">4</span>

    sys.http.prn('</a>');

    if c = max_rows then
        exit;
    end if;

    end loop;

sys.http.p('</div>');

if c = 0 then
    http.p('No recent feedback found');
end if;

end;

```

## APPENDIX M

CODE FOR SOME SECTION OF THE EVENT CALENDAR PAGE



The calendar is a bootstrap calendar plugin that has to be imported in order to use it in the application, after several months of work, I customized it to suit the application.

In the region source, enter the code

```
SELECT  id,
        title,
        all_day,
        TO_CHAR(start_date, 'YYYY-MM-DD"T"HH24:MI:SS') AS start_date,
        TO_CHAR(end_date, 'YYYY-MM-DD"T"HH24:MI:SS') AS end_date,

        'https://localhost:8181/apex/f?p=109:34:&APP_SESSION.::::P34_ID:'||id
        AS url,

        event_color

FROM EVENTS
```

But you need to create a table of events with these columns.

## REFERENCES

- [1] Scott Spendolini, *Expert Oracle Application Express Security*. New York: Manning, 2013.
- [2] Jennifer Kreie, "Using APEX to build a working prototype application based on a database in Oracle," Nov 2012. [Online]. Available: <http://web.nmsu.edu/~jkreie/oracle/DatabaseTopics.html> [Accessed: March. 10, 2014].
- [3] Marcel Van Der Plas and Michel Van Zoest , *Oracle APEX Cookbook* . 2<sup>nd</sup> ed. Birmingham: Packt Published Ltd, 2013.
- [4] Oracle Help Center, .(2014). Database Security Guide [Online]. Available: [https://docs.oracle.com/cd/B28359\\_01/network.111/b28531/vpd.htm#DBSEG98222](https://docs.oracle.com/cd/B28359_01/network.111/b28531/vpd.htm#DBSEG98222)
- [5] J.D. Meier, Alex Mackman, Michael Dunner, Srinath Vasireddy, Ray Escamilla and Anandha Murukan, . ( 2003, June). Improving Web Application Security: Threats and Countermeasures Roadmap[Online]. Available: <http://msdn.microsoft.com/en-us/library/ff649874.aspx>
- [6] Oracle, . (2009). Managing Application Security[Online]. Available: [http://docs.oracle.com/cd/E14373\\_01/appdev.32/e11838/sec.htm](http://docs.oracle.com/cd/E14373_01/appdev.32/e11838/sec.htm)
- [7] Oracle, . (2012). About Oracle Application Express Architecture [Online]. Available: <http://www.oracle.com/technetwork/developer-tools/apex/application-express/apex-deploy-installation-1878444.html>
- [8] Burleson Consulting, .(2014). Oracle Row Level Security [Online]. Available:[www.dba-oracle.com/security/row\\_level\\_security.htm](http://www.dba-oracle.com/security/row_level_security.htm)
- [9] Oracle Cloud, .(2013). Oracle Cloud Data Offerings [Online]. Available: <https://cloud.oracle.com/database>