****

**De La Salle University • College of Computer Studies**

Web Applications Development (WEBAPDE)

AY 2014-2015 Term 3

**Documentation for the Final Output**

Name (last name first) : Aquino, Kurt Neil

Choy, Matthew Seaver

Milan, Jan Tristan

Submitted to : Sir Arturo Caronongan

Section : S18

Date of Submission : April 25, 2015

Time of Submission :

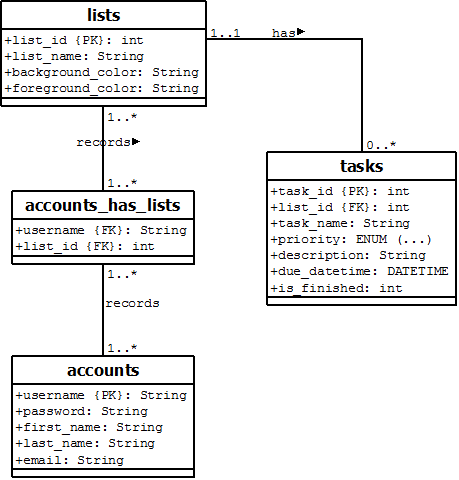
1. **Simon Says: Summary and Marketing**

Simon Says is an online application which keeps track of the user’s daily activities, tasks, events, appointments, etc. in an organized and detailed manner. In today’s fast paced lifestyle, people should be able to manage their time in order for them to prioritize their corresponding objectives for the day. With the application, users will be able to stay up to date with regards to their enlisted tasks and events as reminders and alerts will be sent to their devices via email notifications for as long as they have registered an account in the application. Users can say goodbye to missed deadlines or late appointments as they have the option set the priorities of each task they create as well as select when or how often they would like to receive alerts or reminders prior to their specified due dates.

1. **Technical Report**

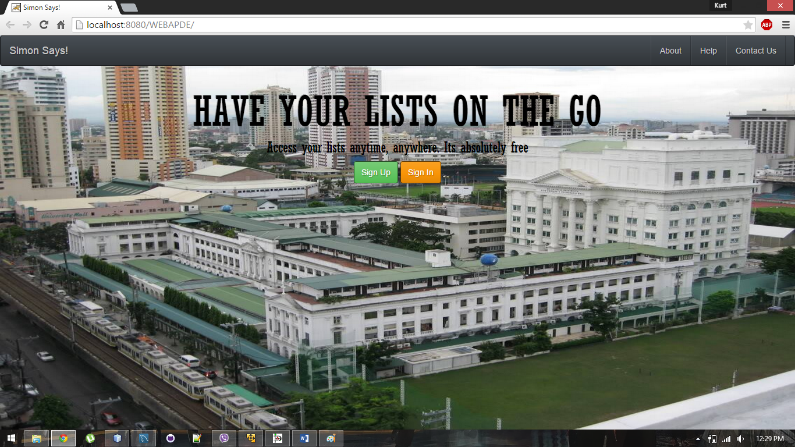
**2.1 Entity Relationship Diagram**

Each [account] is represented by their own unique username as selected by the user. When registering an [account], the user is prompted to set a password and input the email address where they would like to receive notifications from the app. After registration, users will now be able to create their own [lists] by setting its corresponding name, and assigning its own foreground and background colors for easy identification. Once a user has managed to create a [list], the application saves the [list]’s auto generated id with the username of the [account] who made the [list] into the database. The reason for assigning a dedicated table, namely [accounts\_has\_lists], for the list of users and their corresponding todo lists is because users will have the option to share their own lists to other users who use the app as well. After creating a [list], the user can now add [tasks] and set their respective task names, priorities, descriptions, and due dates.

****

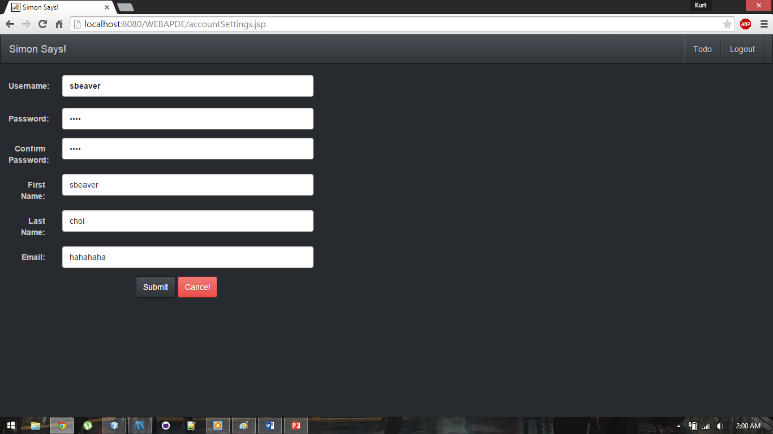
*Figure 1. Entity Relationship Diagram*

* 1. **Servlets and Modules Used**
* **Home Page –** home page of the web application where the user can either login or register an account.



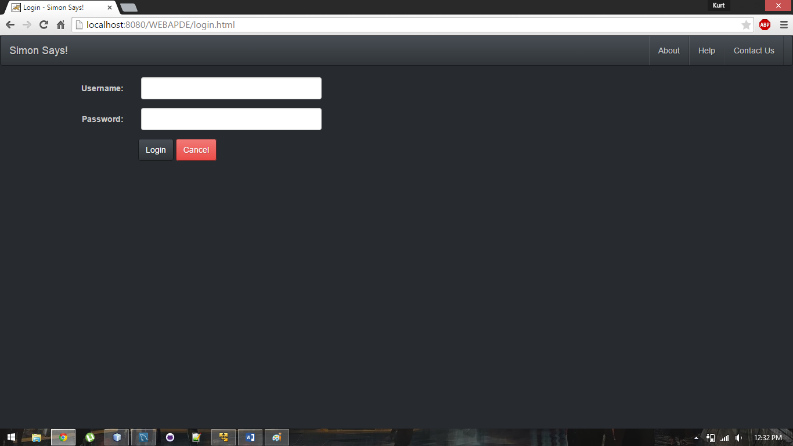
*Figure 2.1 Home Page*

* **Account Management/RegisterServlet/EditAccontServlet –** allows the user to create a new account. Before an account is registered, the system first checks if all of the required fields has been completed, if the username already exists, and if both password and confirmation passwords match. After registration, the user is redirected back to the home page. When logged in, the user has the option to edit his or her account details whenever necessary (except for the account’s username).



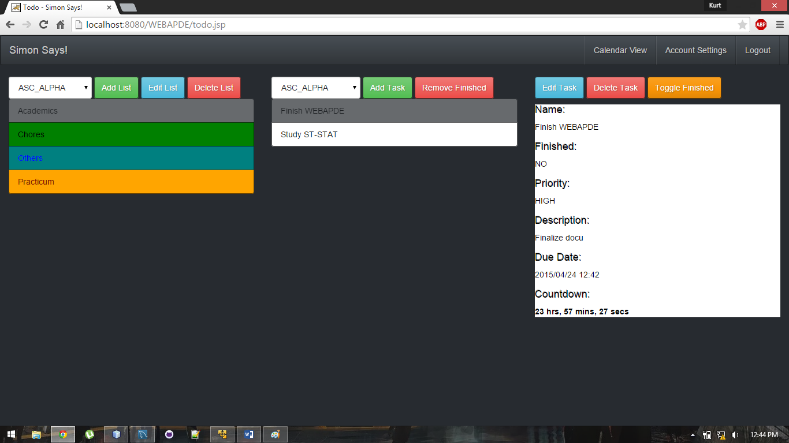
*Figure 2.2 Register/Edit Account Page*

* **LoginServlet –** searches if the input username exists in the application’s database and checks if the input password matches with the one assigned to the account. If the account exists and the username matches, the servlet saves the session with the account’s username as the identifier, and then finally redirects the user to the main page. Otherwise, an error message will pop up and will prompt the user to input a username and password again.



*Figure 2.3 Login Page*

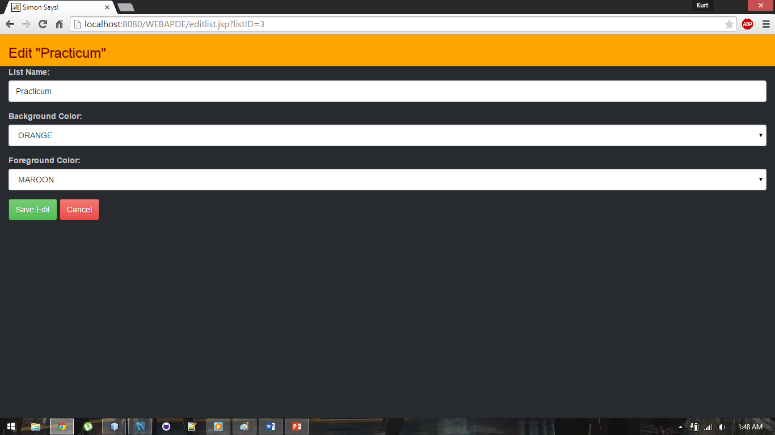
* **LogoutServlet** – given that a user has already login, the logout servlet invalidates the session when logging out and then redirects the user back to the login page.
* **Main Page –** Loads and displays all of the user’s saved todo lists and tasks into the page. Also includes the functions available for managing the user’s lists and tasks.

****

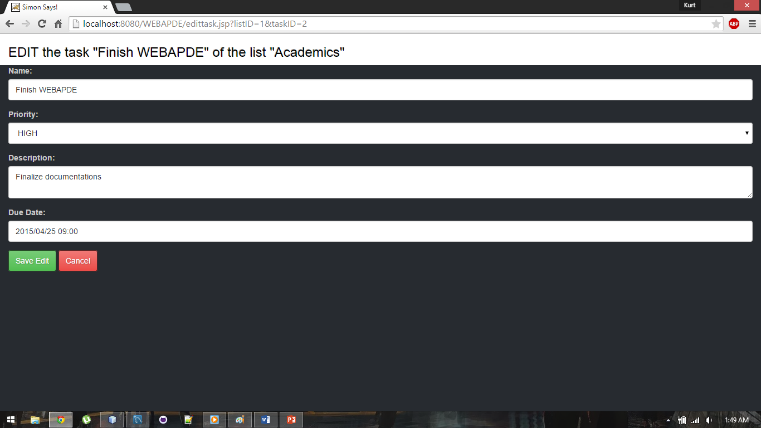
*Figure 2.4 Main Page*

* **List Management/ToDoListServlet –** contains the different functionalities of the application with regards to managing the user’s todo lists. Functionalities include: getList, addList, editList, deleteList, and removeFinishedTasks.

*Figure 2.5 Add/Edit List Page*

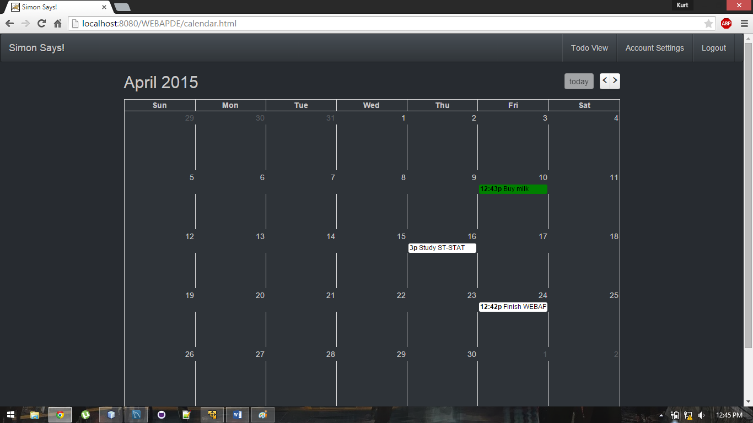


* **Task Management/TaskServlet –** contains the different functionalities of the application with regards to managing the user’s tasks. Functionalities include: getTask, addtask, editTask, finishTask, and deleteTask.



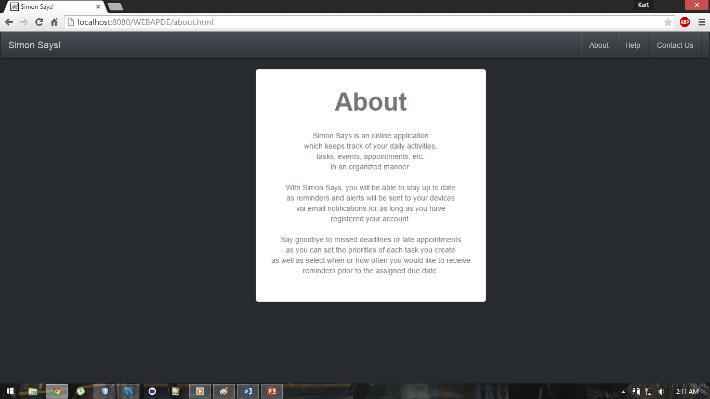
*Figure 2.6 Add/Edit Task Page*

* **Calendar View/CalendarServlet/Calendar API –** loads and displays all of the user’s saved tasks into a calendar view.

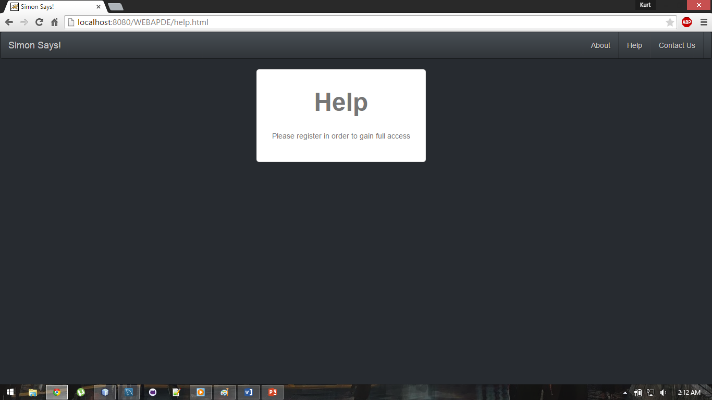


*Figure 2.7 Calendar Page*

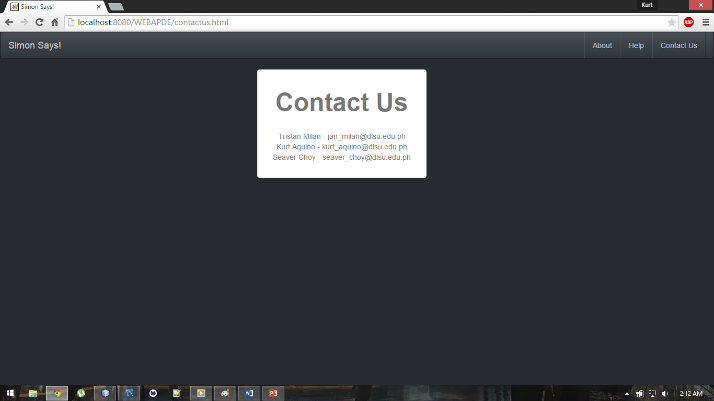
* **About/Help/Contact Pages –** displays other information with regards to the web application and the developers.



*Figure 2.9 About Page*



*Figure 2.10 Help Page*



*Figure 2.11 Contact Page*

* 1. **Web Design/Application Principles Used**
* **HTML/CSS –** mainly used for the overall design and layout of the pages used by the application.
* **Javascript/JQuery –** used for querying and fetching data from the input fields found on specific pages in order to be used for functions. Also used as event handlers to perform specific actions within the web pages.
* **Sessions/Servlets –** sessions were used in our web application to store the information that is desired to persist throughout the user's interaction with web application. To put it simply, sessions were used so that the server-side would remember the state or latest activity the user has done/performed in the web application prior to leaving.
* **Database Connection/Querying –** used for storing, fetching, and manipulating the data needed by the web application to run from its dedicated database.
* **AJAX –** allows our web application to become more interactive as it runs pages dynamically without interfering with the display of the page despite sending/retrieving data from the server in the background.

1. **Self-Evaluation**
   1. **Contributions of Members:**

* **Aquino, Kurt Neil**
* Database Management
* Models
* Query scripts
* HTML pages for the first iteration
* **Choy, Matthew Seaver**
* Debugging the Calendar API to be used for the web application
* Controller
* HTML pages for the first iteration
* **Milan, Jan Tristan**
* Controller
* Servlet Management
* HTML pages for the first iteration
* Converting of HTML pages into JSP (in order to use function calls from the models)
* Applying AJAX to display the list of tasks and todo lists dynamically into the page.
  1. **Further Development and Marketing**

Considering that our web application’s objective is to ensure that the users stay on track when it comes to their appointments, tasks, and deadlines, our team believes that our application would also be best suited for mobile use. Nowadays when most people already have a smartphone at hand, having a dedicated app ported to the phone would be much easier and quicker to use as compared to logging in to the website from a bowser. Accessing and managing your tasks quickly and efficiently would be best suited for a fast paced lifestyle. In order for us to market our app, we should first use it ourselves in order for us to show others how it can really help manage our time and handle our priorities.