# System Architecture for Quarters

Team 6

James Anthony (anthonjb)
Wenqiang Chen (chenw25)
Carolyn Chong (chongce)
Kevin Ly (lyk2)

February 10, 2016

# Contents



# List of Figures



# Revision History

| Date | Comments |
|---|---|
| January 5, 2016 | Created revision 0. |
| January 11, 2016 | Completed revision 0. |
| January 24, 2016 | Updated introduction from past to future tense. |

# 1 Introduction and Overview

This document provides a general overview as to how Quarters will be built. Lists of anticipated and unlikely changes begin the document followed by the design of the system architecture. The system architecture will be designed in a modular manner to support information hiding and is laid out here through the use of diagrams. In the Detailed Design document Quarters' system architecture is decomposed and the design details explained based on the Software Requirements Specifications (SRS) document.

# 2 Anticipated Changes

1. **Design of user interface:** The user interface is expected to change based on feedback from users during usability testing. The interface is expected to change in ways that better support usability principles.

2. **Removal of features:** Some features are expected to be removed based on user feedback. If usability testing indicates that a specific feature would not be utilized then it should be removed.

# 3 Unlikely Changes

1. **Login via social media:** Allows the user to login using accounts from other services such as Facebook, Gmail, Twitter, etc.

2. **Live chat:** A platform for real-time communication between users who are currently logged on to Quarters.

# 4 Decomposition into Components

Figure 1 presents flow of logging into the application.

1. **Landing page** - Presents an overview of the application, outlining some of the main features, and provide the option to sign up and login.

2. **Signup** - Allows user to register new accounts. Once registration is done, the user is directed to the main application.

3. **Login** - Allows user to login with their existing account. Once registration is done, the user is directed to the main application

4. **Session Manager** - Once the user is verified (against the database), the user information will be stored into the local session.

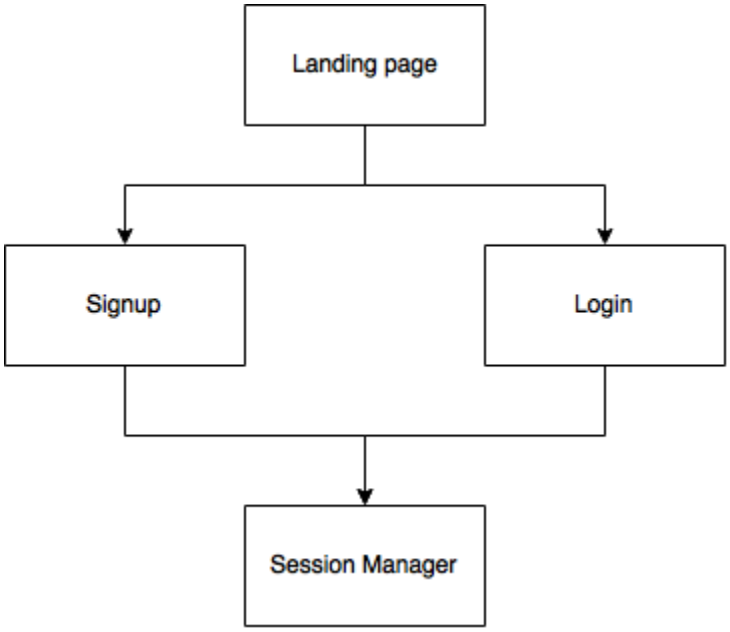


Figure 1: Authentication control flow

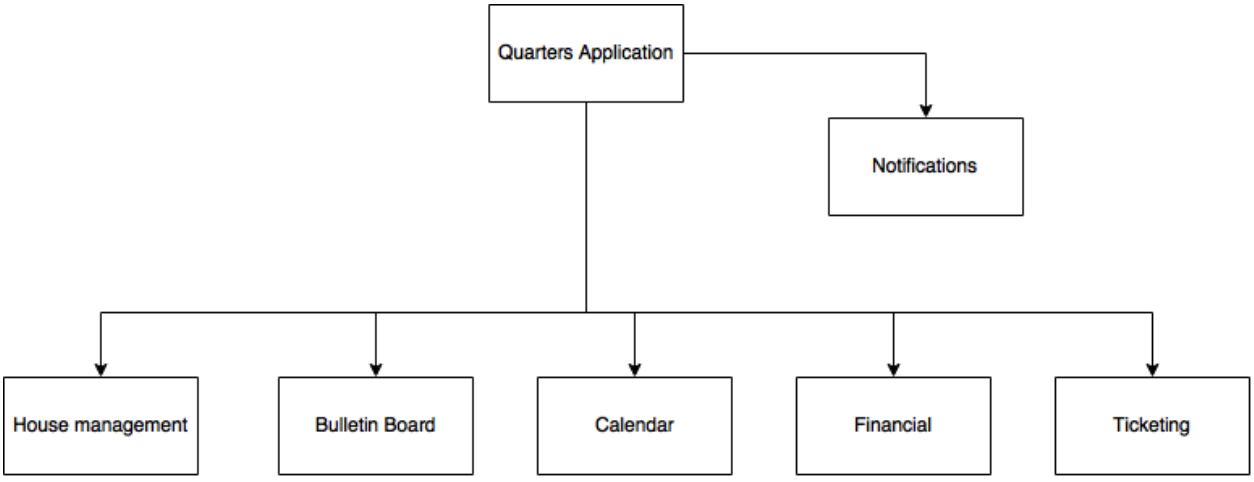


Figure 2: Demonstrates the flow of requirements within the Javascript modules

Figure 2 presents a flow of the Javascript modules

1. **House management** - Handles all activities related to the house. e.g Joining a house, leaving a house, etc.
2. **Bulletin Board** - Handles all activities within the bulletin board module. e.g. e.g. make a new post, delete a post, etc.
3. **Calendar Board** - Handles all updates for calendar events. e.g. make a new event, delete event, etc.
4. **Financial** - Handles all actions for storing financial records. e.g. post a new bill, pay bill, etc.
5. **Ticketing** - Handles all actions for maintenance tickets. e.g. submit a new ticket, close a ticket, etc.
6. **Notification** - Handles all actions for notifications. e.g. post notification.