

Design Document for Quarters

Team 6

James Anthony (anthonjb)

Wenqiang Chen (chenw25)

Carolyn Chong (chongce)

Kevin Ly (lyk2)

January 8, 2016

Contents

1	Introduction and Overview	4
1.1	Document Structure	4
1.2	Design Principles	4
2	Connection between requirements and design	4
3	Anticipated Changes	4
4	Unlikely Changes	4
5	Decomposition into Components	5
6	Uses hierarchy, or control flow diagram, or inheritance graph	5
7	Traceability from requirements to design components	5
8	Traceability for anticipated changes to components	5
9	Error Handling	5
10	User interface elements descriptions	5
10.1	Navigation Flow	5
10.2	Norman's Design Principles	5
10.3	Landing Page	6
10.4	Sign Up	7
10.5	Login	7
10.6	Bulletin Board	7
10.7	House Management	7
10.8	Calendar	7
10.9	Messages	7
10.10	Finances	8
10.11	Maintenance	8
10.12	Notifications	10
11	Overview of key algorithms	10
12	Relational database structure	10
13	Communication protocols specified	11
14	Description of each component, or UI element, or database table	11
15	Development Details	11

16 References	11
---------------	----

List of Figures

1	Quarters UI Navigation Flow Diagram	6
---	---	---

Revision History

Date	Comments
January 5, 2016	Created first draft.

1 Introduction and Overview

1.1 Document Structure

This document provides insight as to how Quarters was built. Design principles are stated followed by a list of anticipated and unlikely changes. The web application's system architecture is then decomposed and the design details explained based on the Software Requirements Specifications (SRS) document.

1.2 Design Principles

TBC. includes a clear statement of what design principle(s) is (are)being used. The web application was designed in a XXX manner. This was to ensure XXX. Decomposition follows the design principle suggested for the design. In many cases the appropriate design principle will be design for change (information hiding). Methodologies include top-down, bottom-up, stepwise refinement, prototyping, modular, or object-oriented.

2 Connection between requirements and design

what design decisions needed to be made to realize the requirements for instance, if there are security NFRs, what decision is made on how to do this password protection?

3 Anticipated Changes

- 1.

4 Unlikely Changes

- 1.

- 5 Decomposition into Components**
- 6 Uses hierarchy, or control flow diagram, or inheritance graph**
- 7 Traceability from requirements to design components**
- 8 Traceability for anticipated changes to components**
- 9 Error Handling**
- 10 User interface elements descriptions**

A description of the user interface design of Quarters is presented here. This section is divided into subsections of the UI navigation flow and the major UI elements. Each UI element is explained with the support Norman's design principles and illustrated with screen images of a mockup. Norman's principles of design are briefly stated to provide some background on the design decisions of Quarters.

10.1 Navigation Flow

See 1. Users are directed to the landing page. From there new users can sign up or returning users can login. After signing up, the new user will be prompted to either join a house or create a house in order to access the rest of the application. After logging in, the returning user will be directed to the main page of the application, called the Bulletin Board. The other pages of the application, including House Management, Calendar, Messages, Finances and Maintenance can all be navigated to from any page using the side navigation bar.

If a user is a member of multiple houses, they can switch between houses via House Management from the top navigation bar.

In-app notifications are visible on the top navigation bar on all pages, thus no navigation is required to access notifications.

The user can access user profile settings and log out via the top navigation bar.

10.2 Norman's Design Principles

Don Norman lists five principles to support software usability. These principles are visibility, affordances, mappings, constraints, feedback, and consistency. Together they guide the design decisions of the UI of Quarters.

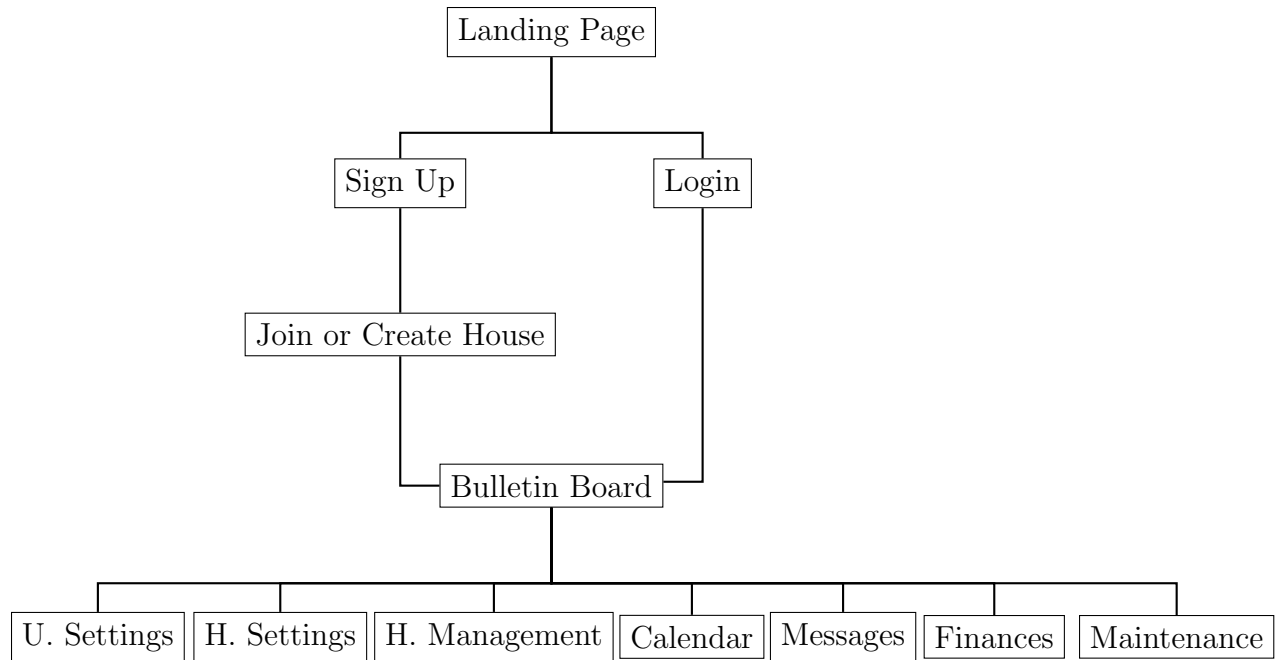


Figure 1: Quarters UI Navigation Flow Diagram

1. Visibility conveys to the user their current state and possible actions.
2. An affordance is a visual attribute of an object or control that helps the user determine how the object or control can be used.
3. A mapping is the relationship between a control and its effects.
4. Constraints limit the ways in which an object or control can be used by the user.
5. Feedback is when the user is informed of the results of their actions and indicates what actions can be taken next.
6. The conceptual model is a physical understanding of an interface or interaction technique based on real-world experience.

10.3 Landing Page

The landing page provides information about the web application's features, and allows the potential user to sign up. There is also a login button for returning users. Both the signup and login buttons are placed in convenient and discoverable locations. The landing page is designed to capture a potential user's interest in the web application by making it appear modern, secure, easy to use, and beneficial through the use of fonts, colors, and layout. A big image covers the landing page that is meant to evoke feelings for a desire of that lifestyle the user could have if they joined Quarters.

10.4 Sign Up

The sign up page is designed to be simple and straightforward to allow for a quick process.

10.5 Login

The login page is designed to be simple and straightforward to allow for a quick process.

10.6 Bulletin Board

The bulletin board (bulletin for convenience) is the main page of the application. From here, every other page can be accessed via the navigation bar. Every user's bulletin is personalized based on their activity. Posts are the focus of this page and are listed in chronological order to be intuitive. Each post is listed with a corresponding icon that symbolizes the type of activity. A large box is positioned at the top of the bulletin to allow users to share content quickly.

Every page of the application has the same layout. There is a top navigation bar, and left navigation bar, and a large space to hold the content of that specific page. The structure of the navigation bars are consistent throughout the application to improve learnability. Fonts, colors and layout are consistent, as well. The whole application experience, from signing up to logging out, is mobile friendly. This means that regardless of screen window size, the user will be able to access all functionality of the application. For example, the navigation bars are collapsed to a toggle menu in smaller screen sizes to ensure the screen real estate is occupied mainly by content that the user will spend the most time looking at.

10.7 House Management

House settings can be accessed from the bulletin. Documents, members and details about the creation of the house are included here. The interface of this section should be simple and uncluttered.

10.8 Calendar

The Calendar resembles the UI of the Google Calendar because it is a widely used calendar that allows for some familiarity. A check list tab is easily accessible to change the visibility of different calendars. Buttons to add events are positioned above the Calendar.

10.9 Messages

A simple chat history of direct messages between two users is displayed here. The messages section is separate from the online users section. An image of the user accompanies each message. Again, a simple, uncluttered interface is important to ensure the user experience is as quick as possible.

10.10 Finances

The main focus of this page is a table displaying all bills.

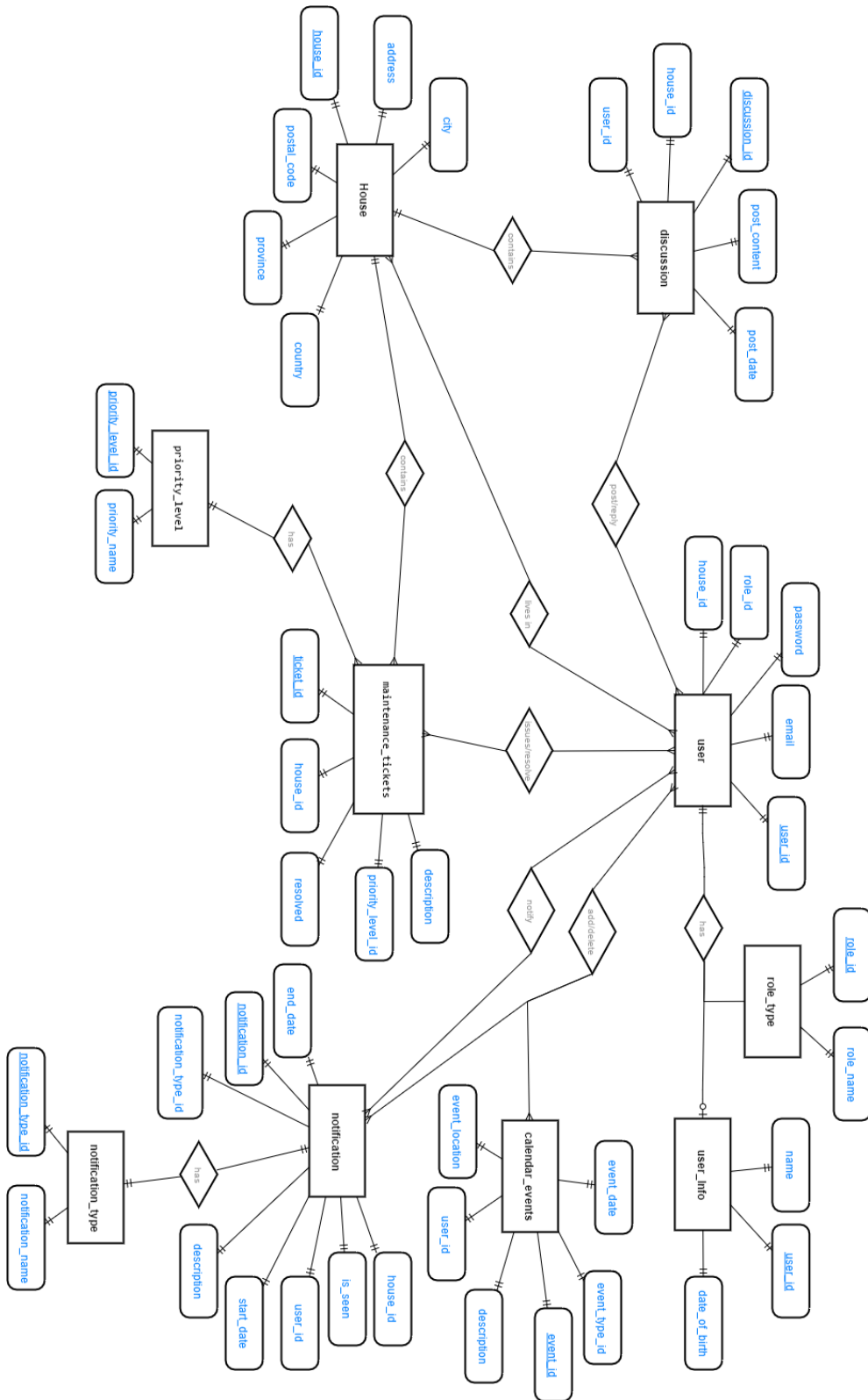
10.11 Maintenance

This section holds a list of maintenance tickets in chronological order. Each ticket is accompanied by a corresponding icon to symbolize the type of ticket. Colors are used to differentiate the priority levels of each ticket. Each ticket is its own horizontal panel.

10.12 Notifications

11 Overview of key algorithms

12 Relational database structure



13 Communication protocols specified

14 Description of each component, or UI element, or database table

15 Development Details

Languages of implementation

- NodeJS [1]
- PostgreSQL [2]
- Jade [3]

Supporting frameworks

- Bootstrap [4]
- ExpressJS [5]

Supporting technology

- Ubuntu Server [6]

16 References

[1] [link](#)

[2] [link](#)