

Design Document for Quarters

Team 6

James Anthony (anthonjb)

Wenqiang Chen (chenw25)

Carolyn Chong (chongce)

Kevin Ly (lyk2)

January 10, 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	User interface elements descriptions	5
7.1	Navigation Flow	5
7.2	Norman's Design Principles	5
7.3	Landing Page	6
7.4	Sign Up	6
7.5	Join or Create House	7
7.6	Login	7
7.7	Navigation bars	7
7.8	Bulletin Board	7
7.9	House Management	7
7.10	Calendar	8
7.11	Messages	8
7.12	Finances	8
7.13	Maintenance	8
7.14	Notifications	8
8	Relational database structure	10
8.1	ER-Diagram	10
8.2	Table Descriptions	11
9	Communication protocols specified	11
10	Development Details	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. **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 increase visibility and are more intuitive to use.
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 .

4 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.

5 Decomposition into Components

6 Uses hierarchy, or control flow diagram, or inheritance graph

7 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 of Norman's design principles and illustrated with screen images from a mockup. Norman's principles of design are briefly stated to provide some background on the design decisions of Quarters.

7.1 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.

7.2 Norman's Design Principles

Don Norman lists six principles to support software usability in his book *The Design of Everyday Things* [1]. These principles are visibility, affordances, mappings, constraints, feedback, and conceptual model. Together they guide the design decisions of the UI of Quarters.

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.

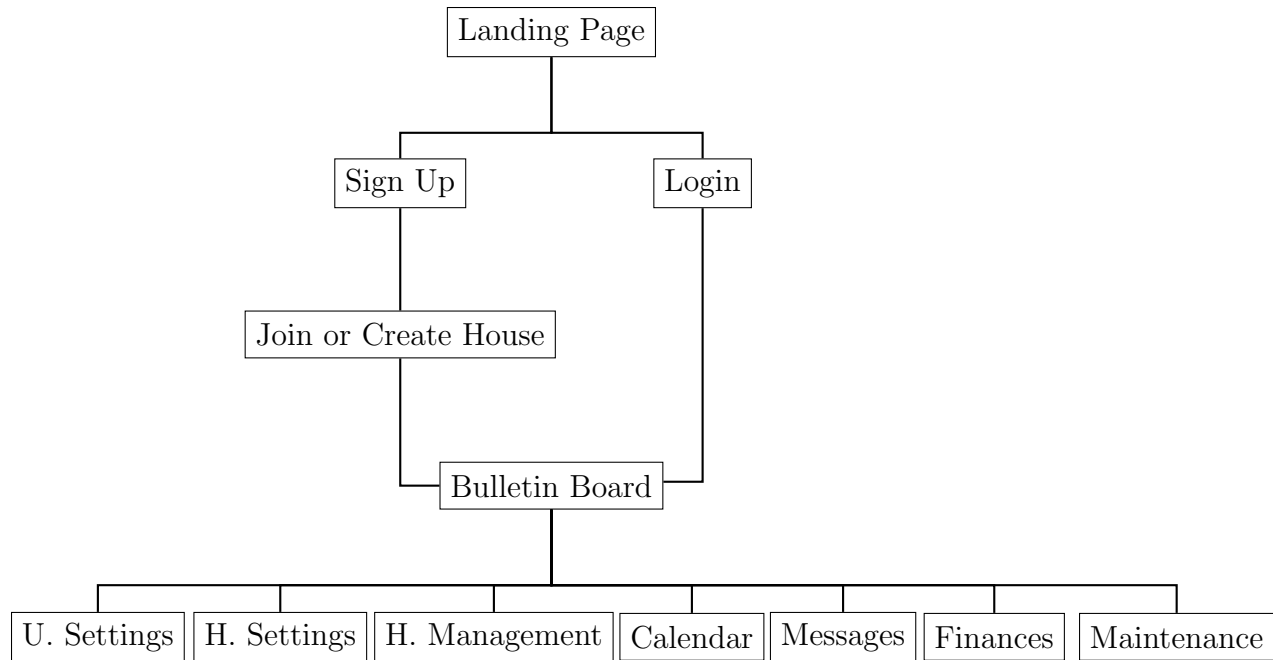


Figure 1: Quarters UI Navigation Flow Diagram

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.

7.3 Landing Page

The purpose of the landing page is to provide information about the web application's features, and to allow the potential user to sign up. There is also a login button for returning users. Both the signup and login buttons are designed to afford clicking and both are placed in convenient and visible 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.

7.4 Sign Up

The sign up page is designed to be simple, straightforward and uncluttered to allow for a quick process. The empty fields afford input. The sign up button is disabled until the user

enters valid input. This feature is an example of a constraint.

7.5 Join or Create House

See House Management below.

7.6 Login

Similar to the sign up page, the login page is designed to be simple, straightforward and uncluttered to allow for a quick process. The empty fields afford input. The login button is disabled until the user enters valid input. This feature is an example of a constraint.

7.7 Navigation bars

Every page of the application has the same layout for consistency. Each page consists of a top navigation bar and left navigation bar, and the remaining space is devoted 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 layouts are consistent, as well. The whole application experience, from signing up to logging out, is mobile friendly. This means that regardless of the screen window size, the user will be able to access all functionality of the application. For example, the navigation bars collapse to a toggle menu in smaller screen sizes to ensure the premium screen space is occupied mainly by content with which the user will spend the most time interacting. The intuitive iconography of the navigation bars clearly indicate there are more options hiding deeper down, and thus, demonstrates strong visibility.

7.8 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 and their house's activity. Posts take up the majority of the screen space and are listed in chronological order to support the conceptual model. Each post is listed with a corresponding icon that symbolizes the type of activity. A large field is positioned at the top of the bulletin to allow users to share content quickly. When the user clicks on the input field it is highlighted to give feedback to the user that the field is active and input is allowed.

7.9 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.

7.10 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.

7.11 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.

7.12 Finances

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

7.13 Maintenance

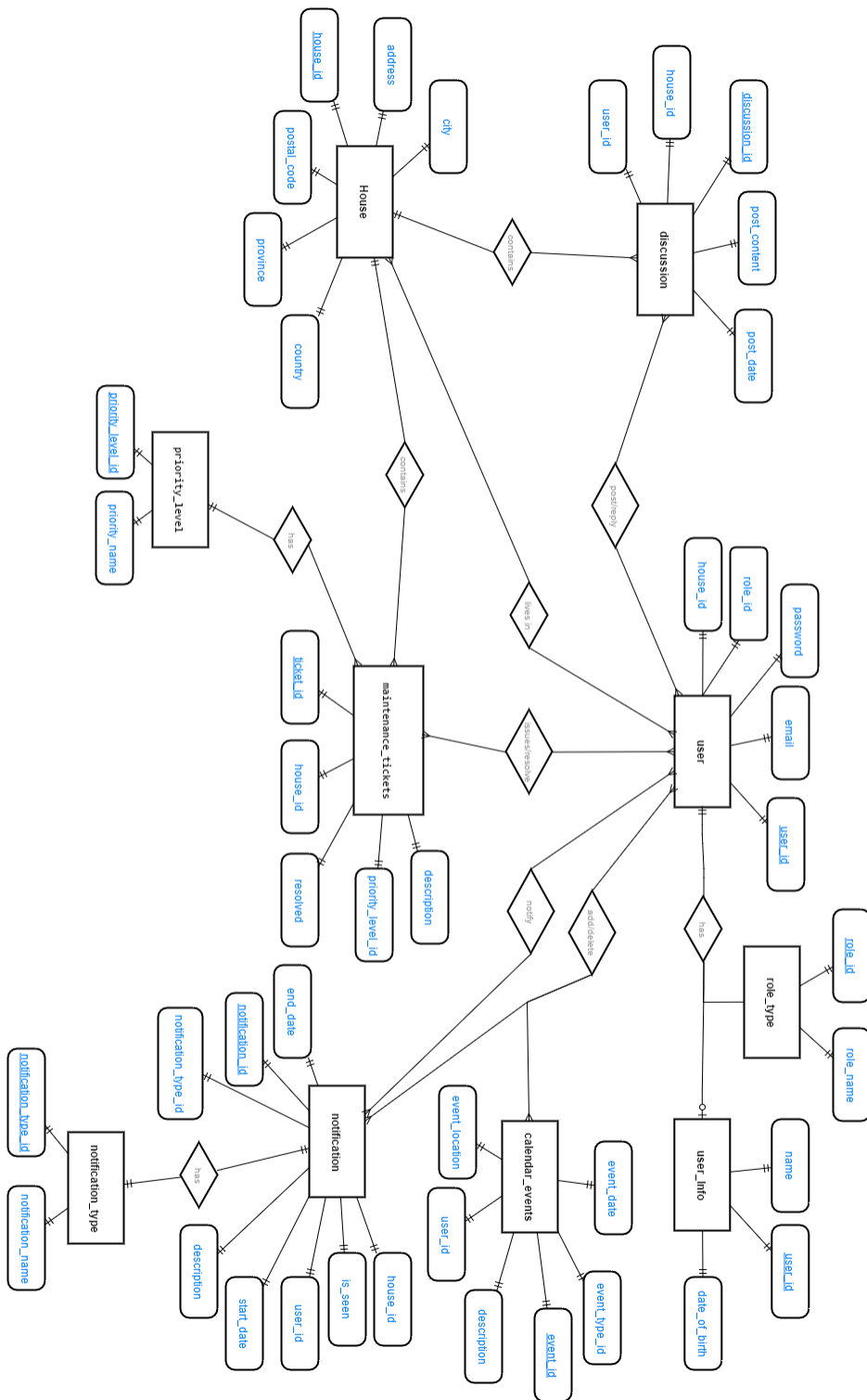
This section holds a list of maintenance tickets in chronological order. Each ticket is accompanied by a corresponding icon to symbolize the type of ticket. Colors are used to differentiate the priority levels of each ticket. Each ticket is its own horizontal panel.

7.14 Notifications

visibility and feedback.

8 Relational database structure

8.1 ER-Diagram



8.2 Table Descriptions

- **user** - Store user credentials(email, password, role type)
- **user_info** - Store basic user informations(name, birthday etc.)
- **house** - Store house informations (address, postal code, etc.)
- **maintenance_tickets** - Store maintenance tickets details
- **discussion** - Store discussion posts
- **priority_level** - Specify priority levels (low, medium, high)
- **role_type** - Specify role types (tenants, landlord)
- **calendar_events** - Store calendar event details
- **notification** - Store notifications for users
- **notification_type** - Specify notification types (discussion, events, tickets, joined house, etc.)

9 Communication protocols specified

10 Development Details

Languages of implementation

- [NodeJS](#)
- [PostgreSQL](#)
- [Jade](#)

Supporting frameworks

- [Bootstrap](#)
- [ExpressJS](#)

Supporting technology

- [Ubuntu Server](#)

References

- [1] Don Norman. *The Design of Everyday Things - Revised and Expanded Edition*. Basic Books, New York, 10-131, 2013.