

# Design Document for Quarters

Team 6

James Anthony (anthonjb)

Wenqiang Chen (chenw25)

Carolyn Chong (chongce)

Kevin Ly (lyk2)

January 7, 2016

# Contents

<b>1</b>	<b>Introduction and Overview</b>	<b>3</b>
1.1	Document Structure . . . . .	3
1.2	Design Principles . . . . .	3
<b>2</b>	<b>Connection between requirements and design</b>	<b>3</b>
<b>3</b>	<b>Anticipated Changes</b>	<b>3</b>
<b>4</b>	<b>Unlikely Changes</b>	<b>3</b>
<b>5</b>	<b>Decomposition into Components</b>	<b>4</b>
<b>6</b>	<b>Uses hierarchy, or control flow diagram, or inheritance graph</b>	<b>4</b>
<b>7</b>	<b>Traceability from requirements to design components</b>	<b>4</b>
<b>8</b>	<b>Traceability for anticipated changes to components</b>	<b>4</b>
<b>9</b>	<b>Error Handling</b>	<b>4</b>
<b>10</b>	<b>User interface elements descriptions</b>	<b>4</b>
<b>11</b>	<b>Overview of key algorithms</b>	<b>4</b>
<b>12</b>	<b>Relational database structure</b>	<b>6</b>
<b>13</b>	<b>Communication protocols specified</b>	<b>7</b>
<b>14</b>	<b>Description of each component, or UI element, or database table</b>	<b>7</b>
<b>15</b>	<b>Development Details</b>	<b>7</b>
<b>16</b>	<b>GanttProject shows a detailed project schedule</b>	<b>7</b>
<b>17</b>	<b>Pert chart shows dependencies</b>	<b>7</b>
<b>18</b>	<b>References</b>	<b>7</b>

# 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. Lastly, error handling is discussed and a project schedule is presented.

## 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.

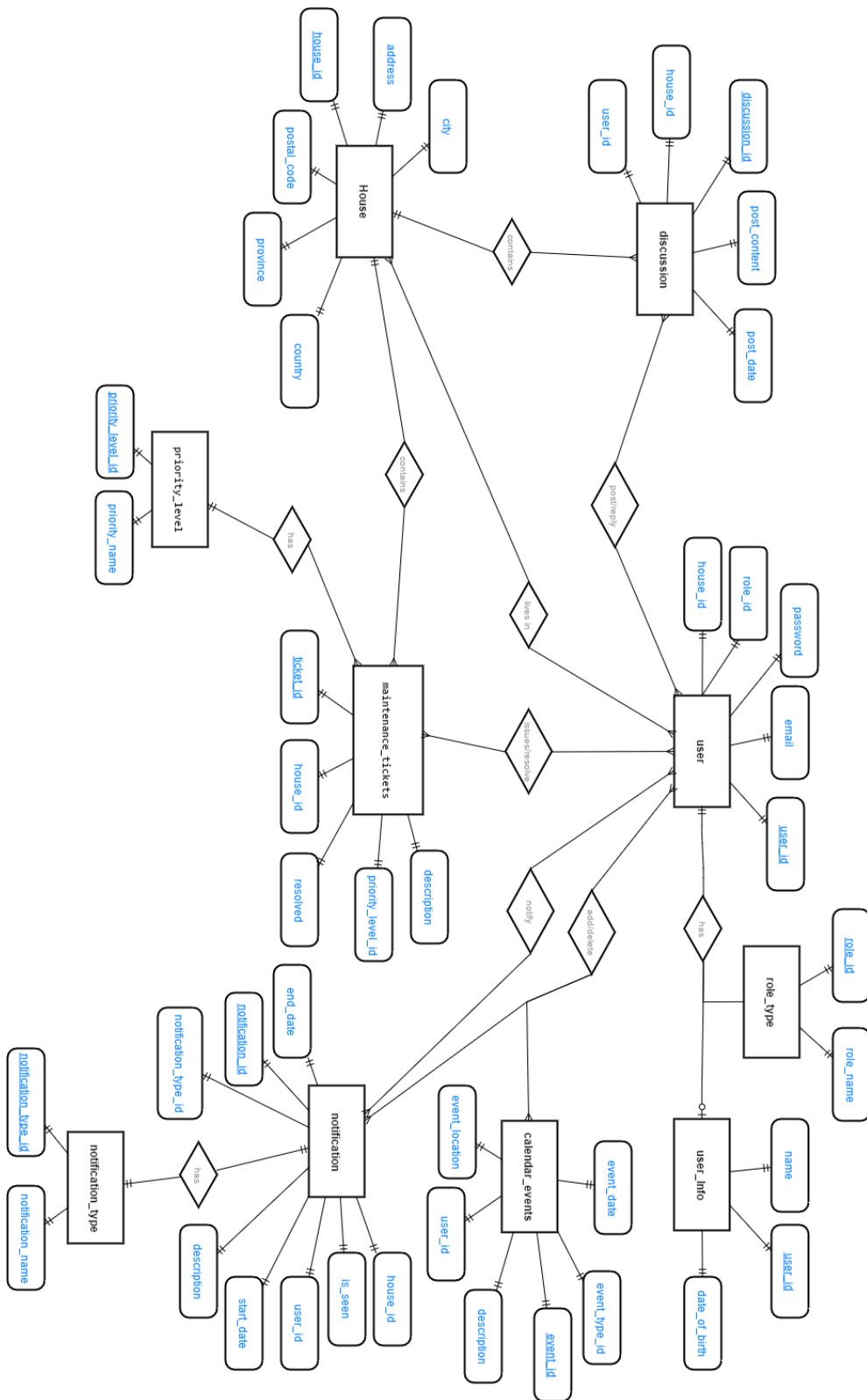
# 4 Unlikely Changes

- 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
- 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 GanttProject shows a detailed project schedule**

## **17 Pert chart shows dependencies**

## **18 References**

[1 ] [link](#)

[2 ] [link](#)