Software Requirements Specifications for Quarters

James Anthony (1135089)
Wenqiang Chen(1155437)
Carolyn Chong (1139105)
Kevin Ly (1144604)

October 12, 2015

Contents

List of Figures

List of Tables

Revision History

Date	Comments
October 9, 2015	Created first draft.

Template

This document makes use of the Volere Template for all of its organization.

1 Project Drivers

1.1 The Purpose of the Project

1.1.1 Project Background

Communication is the exchanging of information between one party to another one. It is an important aspect of people's everyday lives. With the introduction of the Internet and cellular devices people's capacity to communicate has vastly increased, however, this information has hardly centralized. Landlords and their tenants commonly communicate via email or text messages, but a response is not guaranteed or the reply is hard to retrace. [Not with emails —DS] This lack of centralized information may introduce discrepancies between the two parties and hardships upon getting documents and other vital information.

Developing a new software platform to serve as an intermediate between landlords and tenants to act as a centralized hub of information will improve the quality of communication and create a more functional living environment for the occupants.

1.1.2 Project Goal

This platform is designed to handle information between landlords and their tenants, as well as between other tenants. It will efficiently and systemically handle requests from both parties and give a detailed status of the household.

1.2 The Client, the Customer, and Other Stakeholders

1.2.1 The Client

N/A

1.2.2 The Customer

This platform is targeted for tenants and landlords as they will be the primary users of the software. With this implementation, tenants will be able to communicate efficiently with each other and their landlord. This target audience will require an easy to learn service that is robust enough to handle any cases that occur.

1.2.3 Other Stakeholders

There are several other stakeholders than can provide valuable insight to the product. These people and organizations include:

- Business analysts can collect and analyse the potential data.
- Plumbers and electricians can provide information directly to the main stakeholders with respect to solving maintenance issues in the house. [Why only plumbers and electricians? —DS]
- Legal experts can protect the users and developers from unlawful acts. [I don't think this belongs. Can you be more specific? —DS]
- Developers are required to create and test the platform.
- Marketing experts can suggest advertising strategies. [Incredibly vague —DS]

[Where does your supervisor come in? —DS]

1.3 Users of the Product

There are two key users of the product: tenants and landlords.

Tenants are responsible for the following actions:

1. Communicate information to other tenants when needed
2. Complete the chores assigned to them
3. Report maintenance issues in the house via the ticketing system
4. Be punctual with payments
5. Ensure the rules outlined for the house are followed

Landlord are responsible for the following actions:

1. Communicate information regarding the house to the tenants
2. Respond to tenant questions and inquiries
3. Complete maintenance requests within a timely manner
4. Be available when issues arise

2 Project Constraints

2.1 Mandated Constraints

[How are any of these constraints? Constraints are typically given by your client due to their existing technological infrastructure. These are all requirements. If these constraints have been mandated from the beginning, *explain why* —DS]

Solution Constraints

Description: Landlords and tenants shall communicate through a discussion board.
Rationale: Landlords and tenants will not need to communicate via email, text message, or phone.
Fit criterion: All communications conducted via the product's discussion board will be instant and accessible by all members of the group.

Description: Property concerns will be brought to the attention of the landlord by issuing a maintenance request.
Rationale: Tenants will not email, text, or phone their landlord in order to inform them about property related issues.
Fit criterion: Tenants will have the option to pay rent/bills through PayPal via the payment portal. [I think you put this in the wrong place. —DS]

Description: Tenants will be able to make payments through PayPal
Rationale: Landlords will be able to offer a payment method other than post-dated cheques.
Fit criterion: Landlords receive rent/utility payments on time through PayPal from tenants who have agreed to use the service.

Implementation Environment of the Current System

- See off-the-shelf software and collaborative applications
- Server: Intel i3-4430 running Ubuntu Server [If you already have the server, then a solution constraint could be that the software must run on that server —DS]

Partner or Collaborative Applications [Why are any of these constraints? —DS]

- Facebook Login: Provides users with the option to sign up via Facebook.
- Google Sign-In: Provides users with the option to sign up via Google.
- PayPal Payments: Provides users with the option to make payments via PayPal.

Off-the-Shelf Software [Are you going to be using these exclusively? Why are you constrained to using them? —DS]

- PostgreSQL: An object-relational database management system (ORDMBS) with an emphasis on extensibility and on standards-compliance.
- ExpressJS: A NodeJS web application server framework, designated for building single-page, multi-page, and hybrid web applications.
- AngularJS: An open-source web application framework.
- NodeJS: An open-source, cross-platform run-time environment for developing server-side web applications.
- Bootstrap: A free and open-source collection of tools for creating websites and web applications.

Anticipated Workplace Environment

- Home: Website must display properly on desktop and laptop computers.
- Mobile: Website must display properly on mobile browsers.

Schedule Constraints

- Proof of Concept Demonstration, November 16-27

- Revision 0 Demonstration, February 1-12
- Final Demonstration, Exam Period

Budget Constraints
N/A

Enterprise Constraints
N/A

2.2 Naming Conventions and Terminology

Glossary of All Terms

- Landlord: A person who owns or runs a boarding house, inn, or similar establishment.
- Tenant: A person who occupies land or property rented from a landlord.
- House: In the context of this project, a house functions as a set which contains one or more users and stores information about the physical property, the users, and data added by those users.
- Maintenance request: A ticket created by a tenant to inform their landlord of property related maintenance that they are responsible for completing.

2.3 Relevant Facts and Assumptions

Relevant Facts

- Facebook Login is a secure form of account management.
- Google Sign-In is a secure form of account management.
- PayPal is a secure method of making online payments.

Assumptions

- It is assumed that both landlords and tenants will be capable of using web applications for communication, planning, and payments.

3 Functional Requirements

3.1 The Scope of the Work

3.1.1 The Current Situation

There is currently no existing software platform that attempts to simplify and document communication between landlords and tenants. A web application is needed to serve as a centralized management solution that will benefit both types of users. The web application will include document storage, in-app payment, a chore calendar, maintenance ticketing, instant messaging, and a discussion board.

3.1.2 The Context of the Work

See Figure 1.

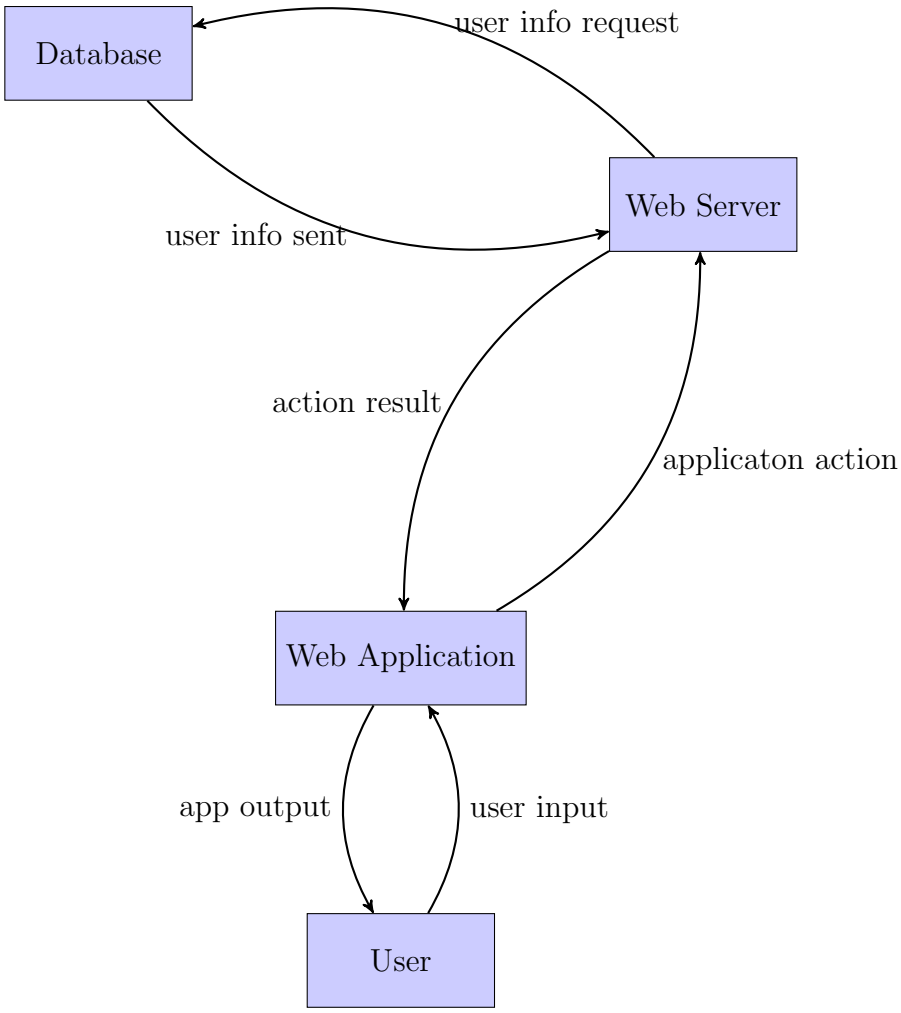


Figure 1: Work Context Diagram

3.1.3 Work Partitioning

See Table 1.

Event Name	**Input and Output**	**Summary**
1. User creates account.	User Name (IN) User ID (IN) User Password (IN) User Email (IN)	User creates an account. The system then adds the account to the database.

2. User deletes account.	User ID (IN) User Password (IN)	User deletes their account. The system then removes the account from the database.
3. User logs in.	User ID (IN) User password (IN)	User logs in to app.
4. User logs out.	User ID (IN)	User logs out from server.
5. User creates house.	User ID (IN) House name (IN)	User inputs the house address into system. The system then saves the house in the database.
6. User joins house.	User ID (IN) House name (IN)	User selects a house to join. The system adds the user to the house.
7. User leaves house.	User ID (IN) House name (IN)	User selects a house to leave. The system removes the user from the house.
8. User uploads file.	User ID (IN) File (IN)	User adds a file to be visible to house. The system then saves the file in the database directory.
9. User submits maintenance request.	User ID (IN) Request (IN) Request (OUT)	User that is tenant submits a maintenance request to be received by landlord.
10. User updates maintenance request.	User ID (IN) Request (IN) Request (OUT)	User that is landlord [Why can't tenants update maintenance requests? —DS] completes a maintenance request.
11. User adds chore.	User ID (IN) Chore (IN) Calendar (OUT)	User submits a chore to be completed and displayed on the Calendar.
12. User initiates chat with other user.	User ID (IN) User ID (IN) Chat (OUT)	User creates chat window with other user.

13. User sends instant message.	User ID (IN) Message (IN) Chat (OUT)	User submits a chore to be completed and displayed on the Calendar. [The summary is wrong —DS]
14. User adds post.	User ID (IN) Post (IN) Discussion Board (OUT)	User adds post to discussion board.
15. User comments on post.	User ID (IN) Comment (IN) Discussion Board (OUT)	User comments on post on discussion board.
16. User transfers funds.	Record of Transaction (OUT)	User transfers funds to another user using third-party software. A record of the transaction is displayed.

Table 1: Work Partitioning

[You mentioned previously that "Electricians and Plumbers" could do things, but they aren't mentioned anywhere in the work partition —DS]

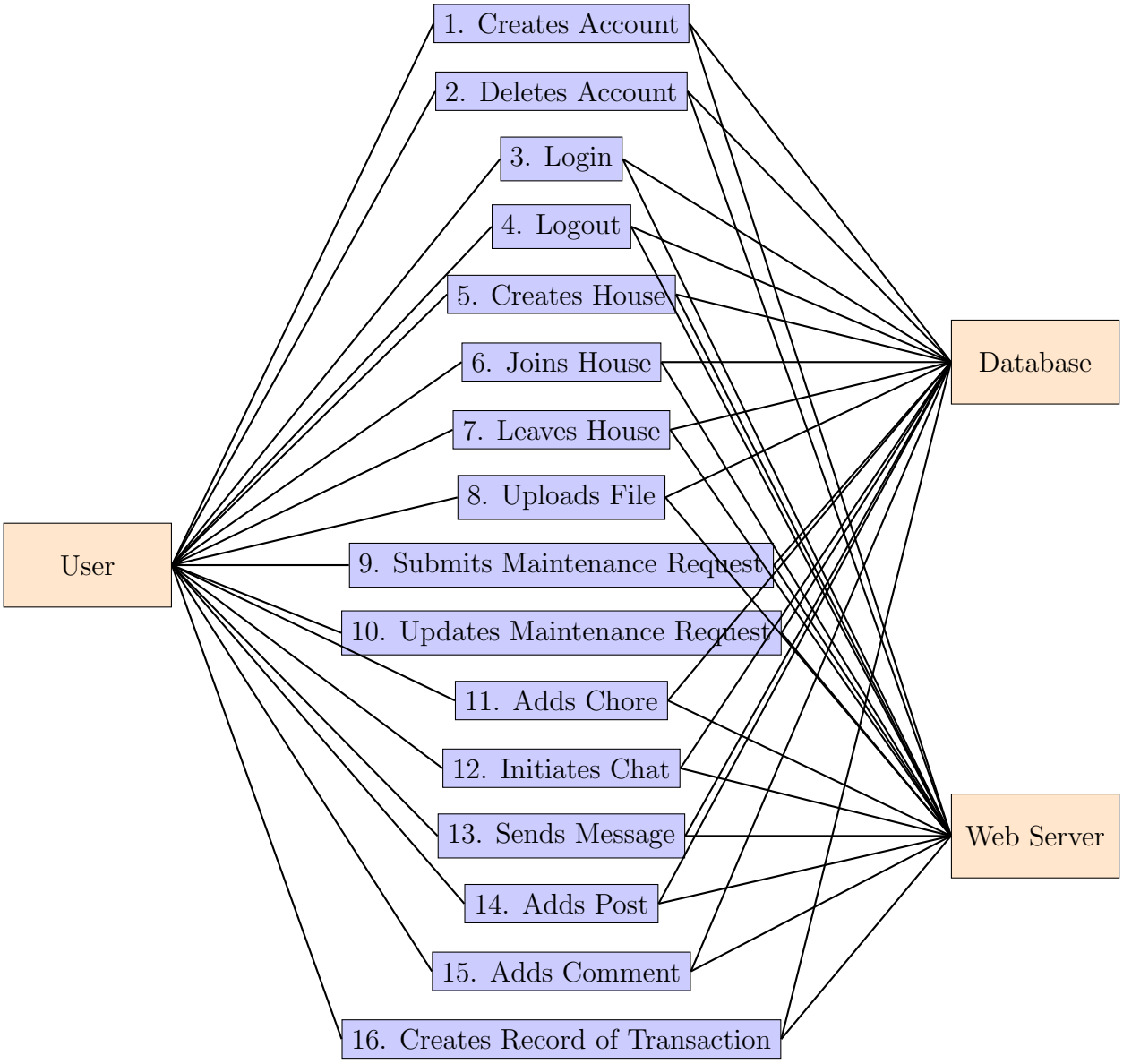
3.2 Business Data Model and Data Dictionary

N/A.

3.3 The Scope of the Product

3.3.1 Product Boundary

See Figure 2.



[You have multiple types of users, thus you should show how they each interact with the system. —DS]

Figure 2: Product Use Case Diagram

3.4 Functional and Data Requirements

3.4.1 Functional and Data Requirements

Requirement #: 1 **Event/Use Case:** 5 **Priority:** 5
Description: User creates a house within the application.
Rationale: To allow users to self-organize and communicate.
Fit Criterion: A house object is created in the database, with the creator as its initial tenant.

Requirement #: 2 **Event/Use Case:** 6 **Priority:** 5
Description: User joins a house.
Rationale: To allow multiple users to join house groups which have already been created.
Fit Criterion: The user is added to the list of tenants associated with the house they are joining.

Requirement #: 3 **Event/Use Case:** 8 **Priority:** 3
Description: User uploads a file.
Rationale: To allow users to share documents.
Fit Criterion: The file that is uploaded is added to the set of files associated with that house.

Requirement #: 4 **Event/Use Case:** 8 **Priority:** 3
Description: User downloads a file.
Rationale: To allow users to retrieve and save documents that have been uploaded by other users in their house.
Fit Criterion: The file selected by the user is downloaded to their machine.

Requirement #: 5 **Event/Use Case:** **Priority:** 2
Description: Users shall be able to specify which other users are able to view content that they post or uploaded.
Rationale: To allow users to share content and communicate privately and keep sensitive data safe.
Fit Criterion: Content shall only be visible to those who the creator has specified are able to view it.

Requirement #: 6 **Event/Use Case:** 9 **Priority:** 4

Description: Tenants shall be able to create maintenance requests.
Rationale: To allow tenants to inform their landlord of property related work that needs to be completed.
Fit Criterion: Landlords shall be able to view a list of current maintenance requests as soon as they are created by their tenants.

Requirement #: 7 **Event/Use Case:** 3 **Priority:** 5
Description: User login.
Rationale: All users should be able to securely login.
Fit Criterion: Upon passing proper credentials, user completes the login process.

Requirement #: 8 **Event/Use Case:** 1 **Priority:** 4
Description: Account registration.
Rationale: Potential users may register an account using Facebook, Google Plus, or registering with Quarters directly.
Fit Criterion: Registering should be a one-step process, with email verification to confirm identity.

Requirement #: 9 **Event/Use Case:** 14,15 **Priority:** 4
Description: Bulletin board.
Rationale: Users are able to post and view information, and comment on other posts to enhance communications.
Fit Criterion: Posts on the bulletin should be sorted chronologically, and viewable by the designated user.

Requirement #: 10 **Event/Use Case:** **Priority:** 5
Description: Multiple resolution compatibility.
Rationale: Users may have different devices with different screen resolutions. The application should be able to support all types.
Fit Criterion: Web platform should support mobile screens and desktop screens, including both horizontal and vertical layouts.

Requirement #: 11 **Event/Use Case:** 6 **Priority:** 5
Description: House management.
Rationale: Administrator should be able to determine which users can access the house.
Fit Criterion: Administrators can add, edit and remove users from a house.

[Who are the administrators? Are they developers of "Quarters" or landlords? —DS]

Requirement #: 12 **Event/Use Case:** 16 **Priority:** 3
Description: Financial platform.
Rationale: Payments between user may be processed within the platform.
Fit Criterion: Payments are secure and logged. Payer and payee will be notified upon payment completion.

Requirement #: 13 **Event/Use Case:** 11 **Priority:** 3
Description: Calendar platform.
Rationale: Events tied to the house can be made in the calendar platform. Chores and other repeating events can be created here.
Fit Criterion: Calendar format should be based off the standardized ICS structure. This format should be deployable to other calendar management software. This platform should support the creation, deletion and modifications to an event.

[How does a landlord end up on a house? —DS] [You have not covered the entirety of your use cases. Where are the rest of your requirements? —DS]

4 Nonfunctional Requirements

4.1 Look and Feel Requirements

4.1.1 Appearance Requirements

The interface of the web application shall be attractive and intuitive for a young adult and adult audience. A sampling of potential users shall, without prompting or enticement, create a login within one week of their first encounter with the application. [The last sentence does not belong. —DS]

4.1.2 Style Requirements

The web application shall appear professional and secure. After their first encounter with the application, 70 percent of potential users shall agree they

feel they can trust the application. [You shouldn't keep copy/pasting examples from the template. They don't necessarily apply —DS]

4.2 Usability and Humanity Requirements

4.2.1 Ease of Use Requirements

The web application shall be easy for young adults and adults to use. The application shall be used by users with no prior training. A casual user should be able to use the application with the same ease of a frequent user. The application shall make the users want to use it. A test panel of current landlords and their tenants shall be able to successfully create a user account and use the application's functions without guidance within their first encounter.

4.2.2 Personalization and Internationalization Requirements

The web application shall be available in the English language (EN-US), use Canadian currency (CAD $), ICS Calendar format, and the metric system. [Why EN-US instead of Canadian or British English? —DS]

4.2.3 Learning Requirements

The web application shall be easy for young adults and adults to learn. The web application shall be constructed so that all of its functionality is apparent upon first encountering it. A brief tour of the web application shall be presented as an option to first time visitors of the site. A test panel of current landlords and their tenants shall be able to successfully create a user account and use the application's functions productively without guidance within their first encounter.

4.2.4 Understandability and Politeness Requirements

The web application shall use symbols, icons, and words that are naturally understandable by the user community. [What does "naturally understandable" mean? —DS]

4.2.5 Accessibility Requirements

The web application shall rely on the web browser's accessibility features to make it available to the disabled. [So you are not implementing any explicit

accessibility options? —DS]

4.3 Performance Requirements

4.3.1 Speed and Latency Requirements

The speed of the web application depends on the speed of the user's operating system and internet connection. [This is not a requirement —DS]

4.3.2 Safety-Critical Requirements

N/A.

4.3.3 Precision or Accuracy Requirements

The web application shall keep accurate time by working in UTC. All monetary amounts shall be accurate to two decimal places.

4.3.4 Reliability and Availability Requirements

The web application shall be available for use 24 hours per day, 365 days per year.

4.3.5 Robustness or Fault-Tolerance Requirements

The web application shall successfully display an error message to the user should an incorrect username/password combination be input, or in the event of one of its features crashing. [There should be error-related functional requirements as well. When the error message occurs, what happens? Does the program stop or can the user continue performing tasks? —DS]

4.3.6 Capacity Requirements

The web application shall cater to 100 simultaneous users.

4.3.7 Scalability or Extensibility Requirements

The web application shall be capable of expanding to nearby cities within two years of its launch. [Will the capacity requirements change with expansion? —DS]

4.3.8 Longevity Requirements

The web application shall be expected to operate as long as there exists a housing rental market.

4.4 Operational and Environmental Requirements

4.4.1 Expected Physical Environment

The web application shall be used by users who may be distracted because they are simultaneously completing and managing several other tasks. [That is not a physical environment —DS]

4.4.2 Requirements for Interfacing with Adjacent Systems

The web application shall work on the last three releases of the five most popular web browsers (Chrome, Firefox, Internet Explorer, Opera, Safari). The web application shall interface with PayPal to handle monetary transactions between users. The web application shall interface with Google Sign-In and Facebook Login to enable users to login with social media accounts. The details of the communication standards/protocols will be outlined in the Design Document after implementation is completed.

4.4.3 Productization Requirements

The web application shall be accessible on the World Wide Web.

4.4.4 Release Requirements

The initial release of the web application will be in February 2016. The next release will be in April 2016. Subsequent releases will be made on an annual basis. [For how long? Can you guarantee it? —DS]

4.5 Maintainability and Support Requirements

4.5.1 Maintenance Requirements

The web application shall be able to be maintained by developers who are not the original developers. [That is not a good requirement. —DS]

4.5.2 Supportability Requirements

N/A. [You may want to include supportability requirements —DS]

4.5.3 Adaptability Requirements

The web application is expected to run on web browsers on mobile phones, tablets and desktop computers.

4.6 Security Requirements

4.6.1 Access Requirements

Only the user has access to edit their own personal stored information and choose what information of their profile is visible to other users. Users have access to view other users' profiles. Only the landlords and tenants belonging to the same property can view the property's group and add content to the property's group.

4.6.2 Integrity Requirements

The web application shall prevent incorrect data from being introduced and protect itself from unwanted attacks by unauthorized users. The web application shall have a back-up of its stored data on an alternate server.

4.6.3 Privacy Requirements

The web application shall make its users aware of its information practices before collecting data from them. The web application shall authorize account registration of users using social media accounts. The web application shall use a third-party interface to store credit card information and perform secure monetary transactions between users.

[Is data stored securely? Is it tamper-proof? —DS]

4.6.4 Audit Requirements

N/A.

4.6.5 Immunity Requirements

N/A.

4.7 Cultural and Political Requirements

4.7.1 Cultural Requirements

N/A.

4.8 Legal Requirements

4.8.1 Compliance Requirements

N/A.

4.8.2 Standards Requirements

N/A.

5 Project Issues

5.1 Open Issues

- Size of the user group is uncertain, therefore a hardware upgrade may be required in the future to accommodate the user.
- User interface has yet to be designed.
- Browser compatibility.
- Methods to store documents, eg. database vs locally on server

5.2 Off the Shelf Solutions

5.2.1 Ready-Made Products

No ready-made products exist with the same functionalities.

5.2.2 Reusable Components

Facebook and Google accounts can be used for sign-in and connecting members in the house.

5.2.3 Products That Can Be Copied

N/A

5.3 New Problems

5.3.1 Effects on the Current Environment

N/A

5.3.2 Effects on the Installed Systems

N/A

5.3.3 Potential User Problems

N/A

5.3.4 Limitations in the anticipated Implementation environment That May Inhibit the New Product

Old browsers are not compatible with the product.

5.3.5 Follow-Up Problems

- User abusing the system.
- User uploading sensitive data.

5.4 Tasks

5.4.1 Project Planning

- Present requirement document to supervisor for feedback.
- Develop prototype for demo purpose

- Refine and develop more features.

5.4.2 Planning of the Development Phases

- Design the database together.
- Design a generalized UI.
- Design back-end of the application.
- Split application into different modules and assign one module to each member for completion.

5.5 Migration to New Product

N/A

5.5.1 Requirements for Migration to the New Product

N/A

5.5.2 Data That has to be Modified or Translated for the New System

N/A

5.6 Risks

- Security/legal issues with online payment.
- Not able to get enough user. ["users" —DS]
- Certain features may not be compatible with the hardware.
- Project becomes too complicated and not able to meet deadlines.

5.7 Costs

- Domain costs and web server costs, if we decide to go live.
- Approximately 7 months of development time.

5.8 User Documentation and Training

5.8.1 User Documentation Requirements

- A help guide will be included.
- FAQ section.
- Tour of the website is shown for first time user.

5.8.2 Training Requirements

No training is required for the user. When they visit for the first time they will be given a tour.

5.9 Waiting Room

The next release will include the following features:

- Separate module for housing advertisements.
- Ability to attach pictures and files in discussion board.
- Notifications either through email or text messages (for urgent events).

5.10 Ideas for Solutions

- PostgreSQL for database management.
- NodeJS and ExpressJS for server side.
- AngularJS and Bootstrap for front end.

[Are these ideas for solutions or constraints? You put them in both places. —DS]