RoomEase Process Description

Alex Vrhel (avrhel), Sid Gorti (sgorti3) Jakob Sunde (jsunde), Omar Alsughayer (oasugher), Cheryl Wang (cwang7), Matthew Mans (mans1626), Weijia Dai (weijid)

Software Toolset

Client Application:

The client application will be developed using Phonegap. Phonegap uses CSS, HTML and Javascript for designing the client application. We decided to use Phonegap, since it allows us to develop one application and compile for multiple platforms. This is important for development, as we want to deploy for Android and IOS to allow nearly all roommates to use the application. Development for IOS and Android has a steep learning curve, and given the tight timeline, it is easier to learn HTML/CSS/JS from scratch than development for IOS or Android.

We will use Materialize for stylizing the application. Materialize provides CSS/HTML/JS implementations for common UI animations and layouts, and therefore we will not have to spend time developing this portion of the UI.

Server Application:

The application will be accessing resources from the internet and therefore a server application will be required. The server application will be developed using Node.JS. Node.JS is relatively simple for new users to learn, and since PHP is relatively difficult to develop with, Node.JS is the best option.

The server application will be hosted using AWS for the reliability and ease of use of the service.

Database:

The application needs to access data from a database and therefore a database is needed. DynamoDB will be used for storing information, since AWS readily supports DynamoDB.

Source Control:

We will use git and host our project on github, since all team member know git and hosting on github allows us to make our source code public.

Task Tracking and Bug Reporting:

We will use Google Docs to keep a list of what tasks are currently being worked on, what tasks are completed, what needs to be completed and what bugs are currently present in the application. A formal task tracking tool is unnecessary for a project of this scope.

Group Dynamics

Alex is serving as the team's Project Manager, and all other group members will share in development, with some specialization. Specifically, Cheryl, Sidd, and Jakob will be working primarily on front end development and UI design, while Matt, Omar, and Weijia will be working more on the back end of the application. These roles were chosen according to our group member's preferences and strengths, although they are flexible and may change as we progress and help is needed in various areas. Disagreements will be handled by taking the issue to the group and deciding democratically, which has proved easy so far due to our open communication channel through Slack. As the PM, Alex is willing to step in to resolve conflicts and mediate issues between group members if necessary. We feel that deciding things as a group when possible is the best method, since everyone has a chance to make their voice heard and be a part of the decision.

Schedule/Timeline

Set */* ***Flexible***

Jan 27 23:59**:**

Everyone: Polished draft of software design spec (for review by TA)
Back-end: AWS & DynamoDB set up

Feb 01: Software design specification

Feb 06 23:59**:**

Back-end: finish shared list feature (does not need to be linked up to 0-feature release)

- *Some communication with the app & DB set up*

Front-end: main pages & ½ of module pages need to be set up (for each main feature)

Feb 08: Zero-feature release

Feb 12 23:59:

Back-end: notifications feed works & reservations
Front-end: UI is linked up with shared list & prepared to integrate with notifications feed

- *Facebook user login*

Feb 16 23:59

Back-end: fridge management feature works
Front-end: full interactivity & integration with backend

Feb 17-18

Testing

Feb 19: Beta Release

Feb 23 23:59

Front end and back end are fully connected, all core functionality complete

Feb 24 - 25

App has been thoroughly tested for Android & iOS devices

Feb 26: Feature-complete release

Feb 26 - March 03

More testing/Early user testing; stretch goals?

March 04: Release candidate & user testing

March 04 - 08

More testing/User testing; stretch goals?

March 08: Final release

JUSTIFICATIONS:
We want half of our app's functionality by the zero feature release because of the short amount of time between ZFR and beta release. Much of our app is UI based and it is critical that we leave enough time for testing, and get features working early enough to get user feedback. For our back end, we want to make sure that our app can communicate with our database by ZFR, since we need to have our back end working and communicating with both our app and our database by the beta release. We want to do at least some early user testing so we can identify early on what some of the potential issues may be.

Risk Summary

1. **Risks to the project caused by inadequate or untestable requirements.**
   Coming up with good and complete requirements is difficult. The functionality of RoomEase are hard to convey precisely and in detail. Moreover, the requirements are hard to communicate effectively for a group of size 7. To reduce this risk, we will try to make our project requirements as detailed as possible during the early stages of development. This also forces us to actively receive feedback from TAs early so we reduce the risk of needing to change our requirements in the future. If we still encounter any problems caused by inadequate requirements, we may adjust our requirements and change our approach so we can deliver our product on time.

2. **Risks of changing software toolkit**
   It is possible that some of the software tools we listed do not work as we expected, or the learning curve associated with the tool is too high. In that case, we will need to find alternative tools. To reduce the risk of changing tools, early on we will experiment with the tools we have chosen so we can see if we need to change the tools we need to use. Even though it is unlikely to happen, if we realise we need to change a tool in the middle of development, we will have to adjust our group dynamics, having one person search for alternatives and others making sure all the other parts of the project are no longer dependent on the old tool. Considering the ramp-up time of getting familiar with a new tool, we will have to change our schedule to accommodate.

3. **Risk of not achieving deadlines**
   Even though we have set milestones for our project, things can go wrong and this makes estimating progress difficult. In order to meet the deadlines, our group will have team meetings frequently to assess our progress. We will also aim to finish things ahead of the specified deadline so we have extra time in case things do not go as planned. When things do go wrong, we will shift the duties of different teammates to help address the issue at hand. If we encounter a major issue that pushes our deadlines extremely far back, we may have to cut features.

4. **Risks of non user friendly UI**
   We are trying our best to make our UI clear and user friendly. To ensure usability, we have had multiple team meetings exclusively for the UI design, and we will continue to do so as the project progresses. Also, we will make sure to continuously show the UI design to our TAs and friends to continuously improve it based on feedback. If we have to modify our UI after it is linked with the backend, we will have to make sure that our changes to the front end do not require heavy changes to the backend. Also we will adjust our group dynamics as needed.

Generally speaking, the feedback from users will the most useful at the beginning of project development. At initial stage, things are not settled yet and the cost of making even drastic changes is comparatively low. Therefore, we will seize our time and ask for as much feedback as possible before our zero-feature release.