Software Requirements Specification

for

HelpMeOut

Version 1.0 approved
Prepared by
LabLabs
3/7/14

Table of Contents

1. Introduction

1.1 Purpose

HelpMeOut is a website that connects people, called Beggers, who are looking for some help doing basic task that they themselves do not want to do. The Choosers, people who help out Beggers, are paid in cash to compensate them for completing their task. The product described in this document is both a website and an android mobile site. Each with different GUI components, which will be discussed later in this document. This document will go into detail about the product and give the reader a detailed overview of what to expect from HelpMeOut.

1.2 Document Conventions

Henceforth, HelpMeOut will be referred to as **HMO** throughout the remainder of this document. The glossary at the end of this document defines any key words (which are in bold) used on our site.

1.3 Intended Audience and Reading Suggestions

This document is intended to give an overview of HMO, to anyone who wants an overview of everything the website entails and what it will bring to the user. The document contains, general information about the product an overview of it's interface, both on web and mobile based applications, a list of its features, and nonfunctional requirements in the product. A table of contents has been provided above for the reader if he or she is looking to read about a specific section about HMO.

1.4 Product Scope

HMO will first start out at the SMU campus and use its student body as a base group. Students will be able to benefit from the product as we test out interest and logistics of the website.We believe the product has a potential to take off among college students so if the product proves to be successful at SMU, HMO will move on and get promoted at other college campus similar to other successful website targeted at college students, such as Facebook. Another goal of this

product is to start at the colleges and then move to the surrounding city from there. This will allow HMO will have a base market and can then spread out (so people asking for help and people looking to help are at a similar ratio).

1.5 References

This document uses various words used on our website that are not only defined in our glossary but also defined in our software lexicon document. The GUI design comes from our paper prototypes and the UI Design Document.

2. Overall Description

2.1 Product Perspective

HMO is designed to be a stand-alone, fully functional web service to help college students outsource simple tasks to their classmates. The application connects people who need help with people who are willing to help, and its economy is driven by small transactions between these two parties. While constricted to SMU students for the alpha testing, HMO is built with the intention of expanding into more spheres of customers.

2.2 Product Functions

- Allow users to post requests for help
    - By category
    - For a specific price
    - To be completed in a specific timeframe
- Allow users to respond to requests by offering help
    - If a user offers help and their offer of help is accepted, they agree to perform the task for the monetary amount specified by the user who originally requested help
    - If a request is declared to be negotiable, a user who offers help may propose a change to the original price of the request

2.3 User Classes and Characteristics

Occasional Users: These users will log on anywhere from once a month to once a week and will be the average user of HMO.

- **Beggars**: these users will occasionally post requests for completion on the feed.
- **Choosers**: these users will occasionally complete tasks for other people.

Frequent Users: frequent users will fuel the HMO marketplace and will help drive traffic to the site and mobile app. These users will effectively create a self-maintaining marketplace where requests are consistently posted and offers are quickly fulfilled. These users will also be more frequently rewarded with in-app currency (Tokens) as well as achievements and badges.

- **Beggars**: these individuals will frequently post requests on a weekly and/or daily basis.
- **Choosers**: these users will help ensure that requests (especially urgent requests) are responded to in a timely manner

Bystanders: these users may never make a request or offer help to another user, but by occasionally visiting the site and/or mobile app, they may develop an interest and generate awareness of HMO through word of mouth.

Hidden Users: if a specific user manages to accumulate a specific amount of negative feedback (TBD) or their requests are flagged as inappropriate, they may be hidden from the system. As a result, any requests made by these users will not be visible to other users and offers for help they make will not be visible by Beggars. By hiding these users, they will not be aware that they have been hidden and will not attempt to make a new account

Administrators: These users will be able to use all of HMO's normal functionality, and in addition will be able to hide/block users, remove/flag jobs, and/or give Token prizes.

2.4 Operating Environment

HMO will primarily be a web application where any user can access the app via a computer with an internet connection. HMO will be accessible through all Operating Systems (Mac, Windows, Linux) and through all web browsers (Chrome, Firefox, Safari, etc.). HMO is also intended to reach Android platforms in app form, but establishment as a website will precede its emergence in the Android market.

2.5 Design and Implementation Constraints

HMO is currently being developed in a very restricted time frame (about 4 months) which will most likely limit the amount of features that we, the developers, will be able to implement in that time. HMO is also being developed on a small webspace, so simulation of an actual, full-blown application with a wide market of users will be limited by our server capacity.

2.6 User Documentation

First time users may want to access the site Lexicon, which will be linked to the HMO web pages, to familiarize themselves with the terminology and the functionality of the HMO site.

2.7 Assumptions and Dependencies

HMO and its developers will assume that its users are running a relatively updated Operating System and Browser configuration. Minimum Recommended configurations are as follows:

Mac: OSX 10.6 or later
Windows: XP SP2 or later
Linux: Ubuntu 12.04+, Debian 7+, Fedora Linux 17

3. External Interface Requirements

3.1 User Interfaces

Web Client:

The web client for HMO will be implemented using HTML, CSS, Javascript, JQuery, and PHP. Every page of the website will display at the top the HMO name & logo (which links to the main page), a section that displays any alerts that the user has not viewed, and a menu bar which contains links to the Home, Login/Account, Post a Job, and Control Panel pages. To the right of the menu bar, the user's name and profile picture will also be displayed, with a link to sign out of their account.

On the bottom of each web page will be links to the following pages: About, Help, User Agreement, Privacy Policy, and our group page (LabLabs). There will be no keyboard shortcuts. Error messages will display when forms are incorrectly or insufficiently filled. Contextual module overlays may occur when the user takes certain actions. Further details of the user interface design can be found in the UI design diagram.

Mobile Client:

The mobile client for HMO will be available on Android devices and will be optimised for a normal screen size, around 720 x 1280. The app will feature a menu bar at the bottom of the screen, divided into 3 sections: Dashboard, My Jobs, and My Account. Pressing the menu button at any time will reveal the following options: View Dashboard, My Jobs, My Account, Sign Out, Help, and About. Error messages will display only if a form is incorrectly or insufficiently filled. Alerts that the user has not viewed will be displayed on the Notifications bar. Selecting an alert from the Notification bar will bring the user to their Dashboard. Further details of the mobile client's user interface design can be found in the wireframes folder in our technical design document folder on the github.

3.2 Hardware Interfaces

Web Client:

The web client can be accessed by any device featuring a web browser.

Mobile Client:

The mobile client interface will be supported only for Android devices. Optimization will be done for a normal screen size, targeting standard mobile phones above tablets or mini-tablets.

3.3 Software Interfaces

Web Application:

The web application will use a 3-tier architecture comprised of an SQL database, a PHP middleware, and a web client.

Mobile Client:

The HMO mobile client will use the same database as the web client. It will use features of the Android OS. In addition, the mobile client will leverage libraries provided by the Android interface. (i.e. notification bar, alerts, etc.) This application will always make HTTP request similar to those found in the PHP middleware of the web application.

3.4 Communications Interfaces

Both the web and mobile clients will communicate with the database via PHP for sending and receiving data, as well as submitting forms. For a user to confirm their email address, an email will be sent to their provided email address that they must respond to before using the application as a user. All messages between the PHP middleware and the web application via a JSON encoding. The HTTP communication standard will be used. Any private data will be hashed via built-in hashing algorithms such as bcrypt.

4. System Features

This website has two main features for it’s users:

4.1 Posting a Job

4.1.1 Description and Priority

This is one of two main features on this site. Posting a job makes the user a “beggar”. Beggars are willing to pay in-app currency to have a job they don’t want to do done for them by a “chooser”. The cost for them is determined by how much they are willing to pay. The benefit is it creates an in website economy driven by the need to have jobs done and to do jobs to get more in app currency. The risk of course is that someone accepts the job and then never does it. The

app allows the beggar to rate the person low. Keeping that misbehaving user for looking good to other members of the community and possibly stopping that behavior.

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

The user will first select his job type. He will give a brief description and set a price and timeframe he needs for the project. Then the user, now called a beggar, will wait for someone to pick up his job. After someone picks his job all he has to do is sit back and wait for a chooser to pick his job and perform the task.

4.2 Performing a Job

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

The other main feature on the site is performing a job. The user performing the job is known as the chooser. This completes the website economy of user paying and user doing the job. Creating a cycle of money, unless the user wants to cash out, and making sure there is a constant stream of jobs. There is no risk on this side as the user's job will be performed and the cash, tokens, will be exchanged through website. Making sure the chooser gets paid for his job.

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

He/She goes through a feed of local jobs and picks the one they want to do. The jobs are displayed to the user by type and how much they pay. After selecting the job the beggar will accept the job offer and the chooser will go out and perform the task. After performing the task both users will select the job is done and go on their way.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Mac: OSX 10.6 or later
Windows: XP SP2 or later
Linux: Ubuntu 12.04+, Debian 7+, Fedora Linux 17

5.2 Safety Requirements

By creating an account, all users of HelpMeOut (HMO) verify that they have read and agreed to the "Terms of Use" (TOU).

AGE: By agreeing to the TOU, users verify that they are 18 years or older. The developers of HMO are not responsible for any unlawful use of HMO services by a minor or the results of such use.

MODERATION: The developers of HMO maintain the right to block any request for help or deny any offer of help for any reason whatsoever. The developers of HMO also maintain the right to prevent any user from accessing HMO services for any reason WITHOUT notification as long as such users do not incur any costs as a result of being denied from using HMO services.

RIGHT TO DENY SERVICE: The developers of HMO maintain the right to deny service to ANY USER at ANY TIME for ANY REASON.

FEES: The use of HMO service is not free to all users, however, the developers of HMO maintain the right to award users “Tokens” for any reason. If a user wishes to continue using HMO services with a token balance of zero tokens, he/she must purchase additional tokens to continue using HMO services.

SALES: Any and all sales made by a user of HMO are final. If a user requests a refund, it is up to the sole discretion of HMO developers to award refunds to users.

5.3 Security Requirements

Since HelpMeOut has a money component to it, the website will be featuring user logins/hash passwords and your traditional security components. As of right now more layers of protection are being discussed and will be added at a later date.

5.4 Software Quality Attributes

Every design, feature and business move put into this website is for the ease of the user.The website is being built with an android mobile client in mind. That means our website will be extremely portable, thus allowing users to reuse and reap the benefits of the site whenever they want to. These frequent users will even be rewarded free tokens The website is going to be aimed at key target markets while starting out which will allow a large user base of **choosers** to help **beggars** and vica versa. The website UI is also built with an ease of use in mind and takes design inspiration from various popular mobile apps on the market.

5.5 Business Rules

Only people who make accounts can be **beggars** and **choosers.** Tokens are used as the websites currency (which are bought with real life money) and will help make sure fake account or people purposefully trying to mess with people will ruin the system. Frequent abusers will be punished for their actions while frequent posters or job fulfillers will be rewarded. Further

encouraging good behavior among the community. Web administrators can also ban bad accounts if an account continues to act out of line. The entire goal of this website is to make the user job easy and not have him worry, so that next time he needs a task done, he comes straight back to us.

6. Other Requirements

Appendix A: Glossary

Beggar: a user who posts a request for help using HMO

Chooser: a user who responds to a request for help by offering help

Tokens: In app currency that will be used to request jobs. It's how we transfer payments from the beggar to the chooser while being able to take a small cut to keep the website functioning.