# CMSC 345

### Software Design and Development

# (Adapted from Susan Mitchell)

# System Requirements Specification Template

General Instructions

1. Provide a cover page that includes the document name, product name, customer name, team name, team member names, and the current date.
2. Number the pages of the document.
3. Number and label all figures. Refer to the figures by number in the text.
4. All sections should have an introductory sentence or two.
5. Do not use vague words and phrases such as may, might, could, possibly, should, assumed to be, some, a little, and a lot. Use strong, definite words and phrases such as shall, will, will not, can, and cannot.
6. Watch your spelling, punctuation, and grammar. It is a reflection on your professionalism.

Be sure that your document is

* Complete - No information is missing
* Clear - Every sentence's meaning must be clear to all parties
* Consistent – The writing style and notation is consistent throughout the document and the document does not contradict itself
* Verifiable - All facts stated are verifiable

Remember that you are required to do a peer review of this document.

When you think you are done with the SRS, ask yourself, "Could someone who was not part of the development of this SRS write the corresponding System Design Document?"

Handyman Finder

System Requirements Specification

#### Table of Contents

Page

## 1. Introduction

* 1. Purpose of This Document
  2. References
  3. Purpose of the Product
  4. Product Scope

2. Functional Requirements

3. Non-Functional Requirements

4. User Interface

5. Deliverables

6. Open Issues

## Appendix A – Agreement Between Customer and Contractor

## 1. Introduction

1.1 Purpose of This Document

This document provides an overview of the Handyman Finder’s functional and non-functional requirements for use in designing and later implementing the product. To be used by both the development team as well as the customer to verify that all requirements are known and being met as part of the design.

* 1. References

W3Schools – [www.w3schools.com](http://www.w3schools.com) – Used as a resource for learning HTML and CSS code for the development team

* 1. Purpose of the Product

This product aims to help normal people find household maintenance services in a quick and convenient way that is also backed up by reviews done by users and businesses. The product allows users to ask for a specific service at a certain price, then connects them with a corresponding business.

* 1. Product Scope

Please see the Functional Requirements for the Use Case Diagrams representing the following numbered arrows.

Users

(All)

10. Change Account Info

9. View History

1. Register or 2. Login

16. Freeze or 15. Unfreeze

8. Admin Access

14. Complaint Review

4. Search for, 5. Select, and 12. Finalize Jobs

Communicates and stores information during website use

6. Review a Job

17. Mark Job Complete

13. File Complaint

7. Contact an Admin

18. View Job Listings

3. Create Job

11. Job Approval

Customers and Workers

Customers

Administrators

Workers

Server

(Back End)

2. **Functional Requirements**

|  |  |  |
| --- | --- | --- |
| **Number** | 1 | |
| Name | User registers on the website | |
| **Summary** | User creates an ID and password which allows them to log in | |
| **Priority** | 5 (Highest) | |
| **Preconditions** | User is connected to the website | |
| **Postconditions** | User’s chosen ID is registered on the web server with the associated password | |
| **Primary Actor** | User | |
| **Secondary Actors** | Server | |
| **Trigger** | User clicks the register button | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | User clicks register |
|  | 2 | User is prompted to create an ID and password, and select account type (Worker or Customer) |
|  | 3 | User creates ID and password with an email account |
|  | 4 | ID, Password, email, and account type are sent to the server for registration |
|  | 5 | User is now registered on the server |
| **Extensions** | **Step** | **Branching Action** |
|  | 4a | Error due to ID already being registered; prompt user for a different ID or to login using that ID if it is theirs |
|  | 4b | Password fails to meet requirements: Must have a number and be at least 8 characters long |
|  | 4c | Email is of an invalid format, so the user is prompted for a valid email until a valid one is provided |
| **Open Issues** |  | |

|  |  |  |
| --- | --- | --- |
| **Number** | 2 | |
| Name | User logs in on the website | |
| **Summary** | User enters login ID and password to log in to website | |
| **Priority** | 5 (Highest) | |
| **Preconditions** | User is connected to the website | |
| **Postconditions** | User is logged in on the website | |
| **Primary Actor** | User | |
| **Secondary Actors** | Server | |
| **Trigger** | User clicks the log in button | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | User clicks log in button |
|  | 2 | User is prompted to enter an ID and password |
|  | 3 | User enters their ID and password and clicks “ok” |
|  | 4 | ID and password are sent to the server for verification |
|  | 5 | User is now logged in to the server |
| **Extensions** | **Step** | **Branching Action** |
|  | 4a | User entered incorrect login information: is not logged in and is re-prompted for ID and password |
| **Open Issues** |  | |

|  |  |  |
| --- | --- | --- |
| **Number** | 3 | |
| Name | Customer Creates Job | |
| **Summary** | Customer Creates a job which is then viewable on the website | |
| **Priority** | 5 (Highest) | |
| **Preconditions** | Customer is logged in already | |
| **Postconditions** | Job is created and listed on the website when searched for using proper criteria | |
| **Primary Actor** | User (Customer) | |
| **Secondary Actors** | Server | |
| **Trigger** | User clicks the “Create Job” button | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | User clicks “create job” button |
|  | 2 | User is prompted to enter information about the job, such as type of work desired, price, first name, and address |
|  | 3 | User clicks “ok” |
|  | 4 | Job and information is sent to the server |
|  | 5 | Server lists job for workers to find |
| **Extensions** | **Step** | **Branching Action** |
|  | 4a | User fails to enter information required for a job, is prompted by the site to finish entering all information |
| **Open Issues** |  | |

|  |  |  |
| --- | --- | --- |
| **Number** | 4 | |
| Name | Worker Searches for Job | |
| **Summary** | Worker enters search criteria and is shown jobs matching those criteria | |
| **Priority** | 5 (Highest) | |
| **Preconditions** | Worker is logged in | |
| **Postconditions** | Jobs matching the input criteria are displayed on the site | |
| **Primary Actor** | User (Worker) | |
| **Secondary Actors** | Server | |
| **Trigger** | Worker clicks the “Search for Jobs” button | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | User clicks “Search for Jobs” button |
|  | 2 | User is prompted to enter search criteria, such as review score, payment amount, closeness of job, type of work |
|  | 3 | User clicks “ok” |
|  | 4 | Criteria are sent to the server, whose job listings are searched for jobs matching the criteria input by User |
|  | 5 | Jobs are listed for the User to browse |
| **Extensions** | **Step** | **Branching Action** |
|  | 5a | Input data does not match any listings on the server; User is informed of the result and prompted to widen their search |
| **Open Issues** |  | |

|  |  |  |
| --- | --- | --- |
| **Number** | 5 | |
| Name | Worker Selects a Job | |
| **Summary** | Worker Selects a Job from the Job Search Listing | |
| **Priority** | 5 (Highest) | |
| **Preconditions** | Worker is logged in and has made a Search using Use Case 4 | |
| **Postconditions** | The User who created the job is notified that a Worker has requested more information | |
| **Primary Actor** | User (Worker) | |
| **Secondary Actors** | Server, User (Customer) | |
| **Trigger** | Worker clicks a button indicating a specific job in their search | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | Worker clicks “More Info” button on a job |
|  | 2 | Server retrieves information for Worker and displays the relative location of the job, its description, and the payment amount |
|  | 3 | User clicks “ok” to confirm their interest in the job |
|  | 4 | Server sends a notification to the User (Customer) who requested the job |
|  | 5 | Jobs are listed for the User to browse |
| **Extensions** | **Step** | **Branching Action** |
|  | 3a | User (Worker) decides instead to go back to search results instead of confirming interest in the job |
| **Open Issues** |  | |

|  |  |  |
| --- | --- | --- |
| **Number** | 6 | |
| Name | Submit a Review | |
| **Summary** | User submits a review of a job that has been completed | |
| **Priority** | 4 | |
| **Preconditions** | The job request has to be completed for a review to be filed | |
| **Postconditions** | The review is submitted and filed on the server | |
| **Primary Actor** | User | |
| **Secondary Actors** | Sever | |
| **Trigger** | User clicks the submit review button next to a job request | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | User clicks the submit a review button |
|  | 2 | User is prompted to fill out a form with a star rating and an optional comment box |
|  | 3 | User fills out the form |
|  | 4 | User hits the submit form button |
|  | 5 | Form is sent to the server for filing |
|  | 6 | User has submitted a review |
| **Extensions** | **Step** | **Branching Action** |
|  | 1a | User fails to enter the required star rating: prompt user that the star rating has to be filled in before submitting |
| **Open Issues** |  | |

|  |  |  |
| --- | --- | --- |
| **Number** | 7 | |
| Name | Contacting an Admin | |
| **Summary** | User sends a message/complaint to the Admins | |
| **Priority** | 4 | |
| **Preconditions** | User must be logged in | |
| **Postconditions** | Admin receives a message from the User | |
| **Primary Actor** | User | |
| **Secondary Actors** | Admin | |
| **Trigger** | User clicks the 'Contact Admin' link/button | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | User clicks contact button |
|  | 2 | User is prompted with a form (Reason for message, comments) |
|  | 3 | User fills out the form |
|  | 4 | User clicks submit |
|  | 5 | Admin receives message |
| **Extensions** | **Step** | **Branching Action** |
|  | 4a | User fails to fill out the required form fields |
| **Open Issues** |  | |

|  |  |  |
| --- | --- | --- |
| **Number** | 8 | |
| Name | Admin Window | |
| **Summary** | Administrator user can view all users' reviews, complaints, and frozen accounts. | |
| **Priority** | 3 | |
| **Preconditions** | User is admin and logged in | |
| **Postconditions** | Admin user can see the admin window and use its features | |
| **Primary Actor** | Admin User | |
| **Secondary Actors** | Server | |
| **Trigger** | Admin user logs in, or loads the homepage while still being logged in | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | Admin user logs in |
|  | 2 | Admin user can then see their homepage |
|  | 3 | Admin user can view user reviews |
|  | 4 | Admin user can see complaints (if any) |
|  | 5 | Admin user can see frozen accounts |
| **Extensions** | **Step** | **Branching Action** |
|  | 4a | If an admin user wants to act on a complaint, they can do so by clicking on it. |
|  | 5a | Admin user can unfreeze an account |
| **Open Issues** |  | |

|  |  |  |
| --- | --- | --- |
| **Number** | 9 | |
| Name | View Transaction History | |
| **Summary** | User views their history of work or requested work | |
| **Priority** | 3 | |
| **Preconditions** | User has requested work in the past or responded to work requests | |
| **Postconditions** | User is given a list of jobs they have been involved in previously | |
| **Primary Actor** | User | |
| **Secondary Actors** | Server, Other Users | |
| **Trigger** | User clicks “View History” | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | Worker clicks “View History” |
|  | 2 | Server retrieves information about previous jobs the user has been involved with, the other users involved in those jobs, and the current User’s role in those jobs |
|  | 3 | User is given a list representing their job history through a display |
| **Extensions** | **Step** | **Branching Action** |
|  | 2a | User may not have a history of jobs. In this case, the server will not return any information, and a page will display instead explaining the user does not have a history of jobs. |
| **Open Issues** |  | |

|  |  |  |
| --- | --- | --- |
| **Number** | 10 | |
| Name | Change account info | |
| **Summary** | User/Admin changes information on their account (picture, address, email, password) | |
| **Priority** | 4 | |
| **Preconditions** | User is logged in | |
| **Postconditions** | New user account information shows on the webpage | |
| **Primary Actor** | User/Admin | |
| **Secondary Actors** | Server | |
| **Trigger** | User clicks the “Change Account info” button/link | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | User clicks the Change info button |
|  | 2 | User is prompted with changeable account information fields |
|  | 3 | User makes the desired changes |
|  | 4 | User clicks the Submit button |
|  | 5 | Updated information is pushed to the server |
| **Extensions** | **Step** | **Branching Action** |
|  | 4a | User enters malformed information and must fix it |
| **Open Issues** |  | |

|  |  |  |
| --- | --- | --- |
| **Number** | 11 | |
| Name | Job Approval | |
| **Summary** | Customer approves a job acceptance for the worker | |
| **Priority** | 5 (Highest) | |
| **Preconditions** | Worker accepted a job that was requested by the customer | |
| **Postconditions** | Job information is sent to the worker upon approval from the customer | |
| **Primary Actor** | Customer | |
| **Secondary Actors** | Worker, Server | |
| **Trigger** | Customer clicks the approve button on their job | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | Customer clicks the approve job button next to a job request |
|  | 2 | Acceptance of the job is sent to the server |
|  | 3 | Sever sends job information to the worker |
|  | 4 | Worker receives information on the job from server |
|  | 5 | Worker can then go complete the job |
| **Extensions** | **Step** | **Branching Action** |
|  | 1a | Customer does not click the approve button. |
| **Open Issues** |  | |

|  |  |  |
| --- | --- | --- |
| **Number** | 12 | |
| Name | Job Finalization | |
| **Summary** | After approval by customer, worker receives the full address and contact information | |
| **Priority** | 5 | |
| **Preconditions** | User is logged in, applied to work for a job, and the other user has accepted the work. | |
| **Postconditions** | Working user can view their client's address and contact information | |
| **Primary Actor** | User | |
| **Secondary Actors** | Server | |
| **Trigger** | Job-having user clicks an “Accept Work” button | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | Worker receives notification of the client's response |
|  | 2 | Worker can then act on the job (start it or message user for more information, or ignore it if denied) |
| **Extensions** | **Step** | **Branching Action** |
|  | 2a | If the client accepts the worker's request, the worker receives notification with the contact information. |
|  | 2b | If the client denies the worker's request, the worker is sent a notification saying that they have been denied with no further information. |
| **Open Issues** |  | |

|  |  |  |
| --- | --- | --- |
| **Number** | 13 | |
| Name | Report a User / File a Complaint | |
| **Summary** | User submits a report of another user or a complaint over a job | |
| **Priority** | 4 | |
| **Preconditions** | User is logged in  User and Worker have completed a job (optional) | |
| **Postconditions** | Admin receives User report | |
| **Primary Actor** | User | |
| **Secondary Actors** | Admin | |
| **Trigger** | User clicks the “File a Complaint” button | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | User clicks the button |
|  | 2 | User fills out the complaint form (Reason for complaint, description of issue) |
|  | 3 | User hits the Submit button |
|  | 4 | Admin receives the complaint message and takes appropriate action |
| **Extensions** | **Step** | **Branching Action** |
|  | 3a | User fails to fill out the form properly and must fix it |
| **Open Issues** |  | |

|  |  |  |
| --- | --- | --- |
| **Number** | 14 | |
| Name | Message/Complaint Review | |
| **Summary** | Admin reviews a complaint | |
| **Priority** | 2 | |
| **Preconditions** | User has submitted a message or complaint to the Admins | |
| **Postconditions** | Admin has viewed the message or complaint | |
| **Primary Actor** | Admin | |
| **Secondary Actors** | Server | |
| **Trigger** | Admin clicks on the view messages & complaints button | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | Admin clicks on the messages & complaints button |
|  | 2 | Server receives request to all messages and complaints |
|  | 3 | Server sends the list of messages and complaints to the admin |
|  | 4 | Admin chooses a message or complaint to view |
|  | 5 | Server receives request for a certain message or complaint |
|  | 6 | Server sends the details of the message or complaint to the Admin |
|  | 7 | Admin can then see the message or complaint that was requested |
| **Extensions** | **Step** | **Branching Action** |
|  |  |  |
| **Open Issues** |  | |

|  |  |  |
| --- | --- | --- |
| **Number** | 15 | |
| Name | Unfreeze Account | |
| **Summary** | An admin user decides to unfreeze a frozen account | |
| **Priority** | 3 | |
| **Preconditions** | Admin user is logged in, there is a frozen account that is desired to be unfrozen | |
| **Postconditions** | Frozen account is now not frozen | |
| **Primary Actor** | Admin User | |
| **Secondary Actors** | Server | |
| **Trigger** | An account is frozen, and an admin decides to unfreeze it | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | Admin clicks on the user in the “Frozen Accounts” section of the admin panel |
|  | 2 | Admin clicks the “Unfreeze” button |
| **Extensions** | **Step** | **Branching Action** |
|  |  |  |
| **Open Issues** |  | |

|  |  |  |
| --- | --- | --- |
| **Number** | 16 | |
| Name | Freeze Account | |
| **Summary** | The user’s account is Frozen until an administrator has reviewed | |
| **Priority** | 3 | |
| **Preconditions** | User’s overall rating is under a 3 (out of 5) and has more than 5 rates by different customers / workers | |
| **Postconditions** | User is no longer able to access the account | |
| **Primary Actor** | User | |
| **Secondary Actors** | Server, Other Users, Admins | |
| **Trigger** | Other user gives poor rating | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | Other user finishes their rating of current user |
|  | 2 | The server calculates the overall rating of the current user |
|  | 3 | The server sets a flag to freeze the account, and the user is no longer able to log in |
| **Extensions** | **Step** | **Branching Action** |
|  | 2a | The user’s rating is over 3 average, or the user has less than 5 overall ratings |
|  | 3a | If the user tries to log in while frozen, they are prompted and informed that the account is under review. |
| **Open Issues** |  | |

|  |  |  |
| --- | --- | --- |
| **Number** | 17 | |
| Name | Setting Job Completion | |
| **Summary** | Users confirm that the job is complete and transaction is finalized | |
| **Priority** | 5 | |
| **Preconditions** | Worker and Customer are matched up, and the job has been approved | |
| **Postconditions** | Worker no longer sees Customer’s credentials and information. An option to rate each other has been prompted | |
| **Primary Actor** | Users | |
| **Secondary Actors** | Server, Other Users | |
| **Trigger** | Both user’s click “Job Complete” | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | Both user’s click “Job Complete” |
|  | 2 | Server removes personal access to the other user’s credentials |
|  | 3 | Rating UseCase is triggered |
| **Extensions** | **Step** | **Branching Action** |
|  | 1a | After one user has selected “Job Complete” the job interaction will auto terminate after 48 hours, in case the other user has neglected selecting the option. |
| **Open Issues** |  | |

|  |  |  |
| --- | --- | --- |
| **Number** | 18 | |
| Name | Customer views their own job listings | |
| **Summary** | Displays active jobs by customer | |
| **Priority** | 3 | |
| **Preconditions** | User is logged in and has active jobs | |
| **Postconditions** | UI shows active jobs and descriptions | |
| **Primary Actor** | User | |
| **Secondary Actors** | Server | |
| **Trigger** | User clicks “View Active Jobs” | |
| **Main Scenario** | **Step** | **Action** |
|  | 1 | Customer clicks “View Active Jobs” |
|  | 2 | Server retrieves information about the Customers active jobs and confirms that the customer indeed has active job listing(s) |
|  | 3 | Server display’s UI showing active job listings and their descriptions |
| **Extensions** | **Step** | **Branching Action** |
|  | 2a | User may not have actives. In this case, the server will not return any information, and a page will display instead explaining the user does not have active jobs. |
| **Open Issues** | |  |

3. **Non-Functional Requirements**

Decide on a standard format for the non-functional requirements (NFRs). Included in the format should be a unique number for each NFR, a priority (1 = lowest, 5 = highest), a clear, concise description, and the test(s) that will be used during system and acceptance testing to verify that the requirement has been met. Make sure that the test numbers correspond to the NFR numbers. Note that you must include a minimum of 10 NFRs specific to product requirements, organizational requirements, and external requirements.

## 4. User Interface

See “User Interface Design Document for *Handyman Finder*.”

5. **Deliverables**

Provide a list of all deliverable items (that is, all artifacts that you will deliver to the customer). This list will include items such as the product itself (What format? Source code? Executable code? Object code?), documentation, and training resources (if any). Specify when (date) and in what format (e.g., hard copy, CD) each will be delivered. A tabular format works well for this section. We will assume that the deliverable items are as follows:

Hard copies of each of the following:

* Systems Requirement Specification
* System Design Document
* User Interface Design Document
* User Manual
* Administrator Manual
* Copies of all Biweekly Status Reports

A CD (or electronic copy in a ZIP file) containing the following:

* Systems Requirement Specification
* System Design Document
* User Interface Design Document
* User Manual
* Administrator Manual
* All source code
* The executable program
* Any other software required for installation and execution of the delivered program.

6. **Open Issues**

Issues that have been raised and do not yet have a conclusion. These issues will be addressed later in the development process.

**Appendix A – Agreement Between Customer and Contractor**

Place on a separate page. Describe what the customer and your team are agreeing to when all sign off on this document. [One paragraph] Include a statement that explains the procedure to be used in case there are future changes to the document. [One paragraph] Provide lines for typed names, signatures, and dates for each team member and the customer. Provide space for customer comments.

**Appendix B – Team Review Sign-off**

Place on a separate page. Provide a brief paragraph stating that all members of the team have reviewed the document and agree on its content and format. Provide lines for typed names, signatures, dates, and comments for each team member. The comment areas are to be used to state any minor points regarding the document that members may not agree with. Note that there cannot be any major points of contention.

**Appendix C – Document Contributions**

Identify how each member contributed to the creation of this document. Include what sections each member worked on and an estimate of the percentage of work they contributed. Remember that each team member must contribute to the writing (includes diagrams) for each document produced.