**CS673 Software Engineering** 

**Team 4 - E-Commerce Pharmacy**

**Project Proposal and Planning**

| Team Member | Role(s) | Signature | Date |
| --- | --- | --- | --- |
| Zahit | QA Leader |  |  |
| Nivan | Implementation Leader |  |  |
| Ryan | Team Leader | *Ryan Burns* | 9/19/2022 |
| Chenfei | Design Leader |  |  |
| Ignacio Moral | Requirement Leader and Configuration Leader |  |  |
| Tsing | Security Leader |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Revision history**

| **Version** | **Author** | **Date** | **Change** |
| --- | --- | --- | --- |
| **1** |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

[Overview](#_g6igqliy7rm)

[Related Work](#_bf21eadgjj29)

[Proposed High level Requirements](#_rgyo4hi9stmq)

[Management Plan](#_ts358bsdtbcv)

[Objectives and Priorities](#_nxeeppkjxgn4)

[Risk Management (need to be updated constantly)](#_tk7yixobah8p)

[Timeline (need to be updated at the end of each iteration)](#_iksrndohvx29)

[Configuration Management Plan](#_j5uvivmxqcsp)

[Tools](#_dzly5b9kz982)

[Deployment Plan if applicable](#_sd8zu6r3jisd)

[Quality Assurance Plan](#_vra5ptwu59qx)

[Metrics](#_vwjduhc9wuah)

[Code Review Process](#_hx3eaiwb8v3m)

[Testing](#_l9xnpmd6hh0y)

[Defect Management](#_5amsh8h9f0c7)

[References](#_pd9euov6m4du)

[Glossary](#_ty3i2nqffhtc)

# Overview

(Please give an overview of your project. It should include the motivation, the purpose and the potential users of the proposed software system, the basic functionality of the proposed software system and the possible technology stack to be used. )

We’re building an e-commerce pharmacy website that sells prescription medications. The motivation is to sell medicine to cut out the middleman directly to sick adults and teens with prescriptions without them having to go to a brick and mortar pharmacy. The basic functionality of the proposed software system would be that a customer will select a medication, present a prescription (optional) along with a legal form of identification (REAL ID or Passport - OR insurance), purchase, and ship the medication to their home address.. Possible technology stack to be used - JS, React, Mongo DB, Express & Node.

# Related Work

(Please describe any similar software systems that you have found through the online research, and the differences between your software and those software systems.)

* Amazon Pharmacy - provides a web-based marketplace for prescriptions with the market price. After selecting a needed medication , a customer can then add their insurance to check the price with insurance and/or they can transfer their Rx to Amazon Pharmacy (by clicking a Big Yellow Button). **What are the differences between our software and this software?** Question for the team… is there anything that our software can offer that these sites can’t?
* Mark Cuban CostPlus Drug Company **What are the differences between our software and this software?** Question for the team… is there anything that our software can offer that these sites can’t?

# Proposed High level Requirements

* 1. Functional Requirements  
     (For each functional requirement, please give a feature title and a brief description using the following format: As (a role), I want to (action), so that (value).)
     1. Essential Features (the core features that you definitely need to finish):

(For each essential features, please give a rough estimation in terms of person hours or an range of person hours - these features MUST be done by end of semester)

* + 1. Desirable Features (the nice features that you really want to have too - some may be completed by end of semester):
    2. Optional Features (additional cool features that you want to have if there is time - ok if optional features weren’t implemented by end of semester):
    3. Existing Features (delete this item if your project starts from scratch):
  1. Nonfunctional Requirements
     1. Security requirements

## Functional Requirements:

* + As a person with a disease, I want to be able to order prescription medicine online so that I don’t need to walk to the store every time.
  + As a person with a need for prescription medicine, I want to be able to order prescriptions online so that I always have them in reserve.
  + As a customer, I want to be able to register my prescription so that I don’t have to have it with me every time I order medicine.
    1. As a user, I want to be able to register with a secure log-in so that I don’t need to register multiple times
  + As a hospital worker, I want users to be registered before buying prescription medicine so that we’re not selling medicine to people that don’t deserve it.
  + As a security expert, I want the software to be key-locked so that people can log-in easily, but others can’t enter.
  + As a hospital worker, I want to have an administration panel so that I can accept or reject the users, depending on the prescription condition.

## Nonfunctional Requirements:

* + Web-based
  + Javascript-Based Application
  + High-level security
  + MongoDB Database
  + Express
  + Online Hosting
  + Accessible on mobile devices
  + Unique Branding
  + Material UI (MUI)

# Management Plan

## Objectives and Priorities (Basically your project goal)

(Please describe your project objectives with highest priority first. Project Goals can include but not limited to complete all proposed (essential) features, deploy the software successfully, the software has no known bugs, maintain high quality, etc).

High priority is to deploy the software successfully with little to no bugs, and maintain high quality.

## Risk Management (need to be updated constantly)

(Please write a summary paragraph about the main risks your group identified and how you plan to manage these risks. Then use the separate google sheet for detailed risk management. The template is provided in the same folder with this file. Please provide the link to the sheet.)

Risk management - If a team member drops the class we will have an emergency meeting to address any shortfalls the loss will cause on Discord. If the MUI doesn’t fit our needs we are aware of other tools that could be implemented. The team has been encouraged to take backup roles to mitigate any potential roadblocks during development.

**Risk Management Sheet Link:**

## Timeline (this section should be filled in iteration 0 and updated at the end of each later iteration)

| Iteration | Functional Requirements(Essential/Desirable/Option) | Tasks (Cross requirements tasks) | Estimated/real person hours |
| --- | --- | --- | --- |
| 1 | Create a homepage for the User / provide a form for prescription submission | Define requirements, consider design structure,  Complete a design  prototype | 56 hrs per person |
| 2 | User should submit a form and upload a file | Integration tests will begin | 56 hrs per person |
| 3 | The program should be complete where the user can search select and upload their prescription | All tests should be passed and there should be no critical bugs | 56 hrs per person |

# Configuration Management Plan

## Tools

(In this project, we will use Git and Github as the version control tools. Please also specify any other tools to be used, e.g. IDE tools, CI/CD tools, container tools, SAST or DAST tools, and any other DevOps tools)

We are going to use VS Code, GitHub, Vercel

* 1. Code Commit Guideline and Git Branching Strategy  
     (Please briefly describe criteria for the code commitment and the branching strategy used, e.g. what are the branches to be used, how the pull request will be used etc. Here is an article to give you some basic knowledge about different git branching strategies: <https://www.flagship.io/git-branching-strategies/>

We will always have a development branch, but we will organize the different functionalities in different branches. For example, if a team is focusing on the back-end development, and they are figuring out the MongoDB database, they will have the “database” branch, so that other branches don’t accidentally push changes that make the development stop working.

## Deployment Plan if applicable

(If you plan to deploy your application (e.g. your web application), briefly describe how you plan to deploy your application).

Vercel will be our deployment of choice, Heroku will be our backup (maybe AWS if that doesn’t work)

# Quality Assurance Plan

## Metrics

(Describe the metrics to be used in the project to measure the quality of your software. Each metric should be measurable and quantifiable. Examples of metrics include product complexity (LOC, # of files, # of classes, # methods, cyclomatic complexity, etc.) , defect rate (# of defect per KLOC), # of test cases, test case pass rate, cost (# of person hours used), # of user stories completed, etc. **The result of these metrics should be reported in the progress report/ iteration summary sheet.**)

| Metric Name | Description |
| --- | --- |
| # of test cases | Total of unit and integration tests. At least one test case for each method |
| test cases pass rate | We aim a %100 passing rate |
| # of resolved bugs | We aim to identify the bugs with ESLint and resolve all the bugs identified |
| # of user stories completed | We aim to complete all user stories related with essential features |
| Test coverage rate | We aim to achieve at least %75 coverage (using Jest) |
| # of critical bugs | We aim to perform static coding analysis with DeepScan.io and to produce 0 critical bugs |

* 1. Coding Standard

(Describe any coding standard to be used)

We will use Prettier(VSCode plugin) + ESLint as a JS coding standard. We may tweak the ESLint rules per our needs.

## Code Review Process

(Everyone should review all documents to be submitted. Here you will mainly describe how the code review will be done. Who will review the code, e.g. design or implementation leader will review all code or team members review each other’s code. Do you use pull requests for the code review? Is there a checklist to help review? What feedback should the reviewer provide?)

Design, Implementation or QA leader will review the code. We will use pull requests for code reviews as well as some tools to standardize commit messages ([commitizen - npm](https://www.npmjs.com/package/commitizen)) and automatize pre-commit testing ([Husky - Git hooks (typicode.github.io)](https://typicode.github.io/husky/#/)) to ease code reviewing.

## Testing

(Both manual testing and automated testing should be considered. Both unit testing and integration testing should be considered. Briefly describe the testing tools/framework to be used, the personnel involved (e.g. the QA leader will focus on the integration testing and each developer will unit test their own code), when and what types of testing will be performed, the testing objectives, etc)

Manual testing and unit testing will be performed by all developers during development and before committing (using React-testing-library). Automated tests will be performed at pre-commit stage (using ESLint + Husky).

Another manual testing will be performed by the QA leader after merging into the develop branch. And integration tests will be written by the QA leader. All tests will be run using Jest. Critical bug testing will be performed on the develop branch by DeepScan.io

## Defect Management

(Describe the tool to be used to manage the defect (e.g github issues). The types of defects to look at. The actions or personnel for defect management. )

We will mainly use Github as a defect management tool. (For mostly bugs, flaws and failed tests). Anyone can report a defect but mostly the Design leader, Implementation leader or QA leader will report a defect and assign it to the related person.

# References

(For more details, please refer to the encounter example in the book or the software version of the documents posted on blackboard. )

# Glossary

(Any acronym used in the document should be explained here)