**M CS673 Software Engineering** 

**Team 3 - Project Name**

**Project Proposal and Planning**

|  |  |  |  |
| --- | --- | --- | --- |
| Team Member | Role(s) | Signature | Date |
| Elijah Curme | Team Leader | *EC* | 02/14/2021 |
| Jay Hwang | Requirement Leader | *JH* | 02/14/2021 |
| Pelin Akbiyik | QA Leader | *PA* | 02/14/2021 |
| Dinara Tiyekbayeva | Configuration Leader | *DT* | 02/14/2021 |
| Chenghao Feng | Security Leader | *CF* | 02/14/2021 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Revision history**

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

[Overview](#_87t9hln2vjz0)

[Related Work](#_mps353x5ezyl)

[Detailed Description](#_fg3z0hpd4q9v)

[Management Plan](#_ds8oyr75pnh1)

[Process Model](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.27177f40uci)

[Risk Management](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.a4oqwntk3mw)

[Monitoring and Controlling Mechanism](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.ywdoc2clc9yt)

[Schedule and deadline](#_tadq5mb0pici)

[Quality Assurance Plan](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.72e1f4uawy2r)

[Metrics](#_b2haznn3yyz2)

[Standard](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.vc72k6dweldv)

[Inspection/Review Process](#_f1c69ifi68h7)

[Testing](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.r5d5mhtlf0kq)

[Defect Management](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.54a4wuncjg1c)

[Process improvement process](#_jhct37ebxxpn)

[Configuration Management Plan](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.hw41vg4ykxen)

[Configuration items and tools](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.bwlb4d4vdox2)

[code commit guidelines](#_yyauft6zr9hw)

[References](https://docs.google.com/document/d/107bVcXdAG-ogRr90PquFB8-aWGvTwSua8pu_O4Kmz6c/edit#heading=h.8mva2050iy7t)

[Glossary](#_ty3i2nqffhtc)

# Overview

Ziczac is an online marketplace in which users may sell to, purchase from, or trade with other users. The motivation for this project is to encourage buy and sell among people in the local area for reducing waste and to become one of a community platform. The potential users of our project are everyone who wants to sell and buy in a local area.

# Related Work

Amazon and ebay are the similar software systems with our project. The key difference is our online market focuses on direct transactions based on nearby regions. The sellers and buyers can communicate with a chat on the web application for deciding price and time for meet up. So that the system does not need money transactions while users are buying, selling and trading.

Similar Web Link:

<https://www.daangn.com/>

<https://www.facebook.com/groups/155067791244691/>

https://offerup.com/

# Proposed High level Requirements

* 1. Functional Requirements
     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)

1. Chat System : As a user, I want to communicate via message (live chat) for direct transaction (15h)
2. Search Engine : As a user , I want to search items, so that only searched items are listed (10h).
3. Sign up: As a new user, I want to create a new account, so that I can access my orders or post an item for sale. (2h)
4. Post item for sale: As a user, I want to post an item for sale, so that it can be purchased by another user. (2h)

Items for sale

User Friendly

Personal Security

Review

Rating

Trading

Location

* + 1. Desirable Features (the nice features that you really want to have too):

Price Sorting

Recommendation

Elimination by rating

Account Deletion / Recovery

* + 1. Optional Features (additional cool features that you want to have if there is time):

Login options with social media (Facebook, Instagram)

* 1. Nonfunctional Requirements
     1. Security requirements

Password strength evaluation

Password Encryption

# Management Plan

## Process Model

Our software process model borrows aspects from many well known models such as Agile, Scrum and DevOps. Similar to Agile, we favor a lightweight process and prioritize working software deliverables above all. We have Scrum meetings often, and seek to use this shared time to clearly define project requirements and individual responsibilities. In this way, technical hurdles can be addressed on an individual basis, and when team members are in a position to help each other with software development tools such as git or jupyter, collaboration is encouraged.

Objectives and Priorities

We are currently working to produce a basic version of our web application as early as possible so the team may have a clear vision of where we are and where we’re going. From this point we plan to ensure our essential requirements are met, before tackling the optional features that may prove more complex to implement.

## Risk Management (need to be updated constantly)

One team member has already dropped the class. We plan to distribute the leftover work to the remaining team members. A remaining risk is that of completed work that is made obsolete by a miscommunication of requirements or change of requirements. This can be best mitigated by clear and consistent communication. Another remaining risk is technological incompetence, which could potentially result in the loss of completed work, in the case of improper use of git/github. This risk can be mitigated by thorough research of powerful commands like git push before putting them into practice.

Risk Management Sheet Link:

https://docs.google.com/spreadsheets/d/1W9W5vSkvMHdwxx6otmp\_xH65uyR1u7baMdwlEZEYIvU/edit#gid=0

## Monitoring and Controlling Tools and Mechanisms

We will use the following tools to facilitate group communication and monitor the project progress.

* + 1. Pivotaltracker Link: https://www.pivotaltracker.com/n/projects/2487103
    2. Slack Link: https://bumetcs673s21.slack.com/archives/C01LW6FL4SU
    3. Github Link: https://github.com/BUMETCS673/BUMETCS673S21T3
    4. Zoom meeting Link:
    5. Weekly meeting time: Sunday

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Iteration | Functional Requirements(E/D/O) | Tasks | Estimated/real person hours | Presentation Recording Link (5-10 minutes) |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |

# 

# Quality Assurance Plan

## Metrics

* + 1. Product metrics: The software is designed to provide platform for users to sell and buy products. The user would create an account, which consists of username, secure password, location and contact information. The user would list the products that they are willing to sell. The user would upload a picture of the product and write the description of the product. Users who are interested in purchasing products would indicate the location and the type of product that they are interested in. The user would be able to see the reviews for each seller and contact the seller that they want. The seller and the buyer would then communicate through the messaging on the website and negotiate the price of the product. Once the user and the buyer agree, they would meet at a location to complete the transaction. Several classes have been created. Account, order, item and password manager. The method for the project is using Python’s pickle module in order to store the data in a file. The credentials on the website will be checked against this file storage. Same method will be used for an inventory of items for sale.

So far, we have two Python files. One to store user data and one to store inventory. We have created 3 classes, which are Account, Order and Item.

Process metrics: Unit test, integration test and system test will be applied to analyze and improve the codes.

Project metrics: Total of five software developers are working on the project. Each software developer is responsible for one of the aspects of the project, however team work plays an important role and each developer participates in all aspects of the project. Responsibility assignments, deadlines and productivity plays an important role. At least once a week, team meeting is conducted to discuss the project. The goal is to be efficient because the team has to accomplish a lot in a short period of time. User friendliness and security plays an important role in this project. The website has to be attractive to get as many user as possible, at the same time it has to be secure in order to protect user information and avoid scams. To achieve this, we put password criteria’s in place and disqualified seller who received points below a certain number. In addition, users can contact the customer service in order to complain about a seller.

Product quality metrics: Consists of defect density and customer satisfaction. Defect density is defect per unit of code. It is calculated as number of defects divided by the number of lines in the code. Our goal is low defect density. Customer satisfaction indicates the product quality. Our goal is to have minimum customer complains and create a user friendly website. To achieve this, we ask users to submit a survey about the website and make improvements based on their answer.

* + 1. Results:

Iteration 0 - TBD

* 1. Standard

We use HTML/CSS/JS for front end and Python Flask for backend.

## Inspection/Review Process

## 

Review will be done mainly by the QA leader, however the entire team will also review. The quality and user friendliness of the software will be reviewed after completing the framework of the software Improvements will be done based on the result. Inspection on the software will be done frequently to make sure there are no defects or logic errors in the code

## Testing:

## Link to the testing document: TBD

QA will conduct the testing mainly, however all the team members will participate. Testing will be conducted each time the code is updated. Each time the test is conducted, the results will be documented to a separate file with date of the testing recorded. Tests will consist of debugging and fixing the errors if there exists any.

## Defect Management

If the software does not serve its purpose, that means it is defected. Debugging would be the tool to manage defect.

# Configuration Management Plan

## Configuration items and tools

Github (.gitignore) for source and configuration codes along with documentation (readme.md and other documents)

Github files structure: TBD

Slack with incorporated github notifications

## Change management and branch management

Branches: Each username has his/her own branch

Github files naming convention:

file\_name\_v\_version.file\_extension

First version starts with 1.0, second version 1.1 and so on

Github folders naming convention:

folder\_name\_letter

where letter represents how often file needs to be backed up

A - once a year

B - once a month

C - once a week

D - daily

## Code commit guidelines

1. Clear git commit message which outlines what changes have been made
2. Commit message has link to testing document that describes what testing was performed on changed code
   1. Integration and deployment plan

Group is thinking on using Heroku.

1. Send an email to all members that you are starting deployment, include description and release version
2. Document details based on the following workflow:

Planned (/Delayed)→ In Progress → Deploy Complete → Post Validation Complete

# References

Risk Managing Sheet: <https://docs.google.com/spreadsheets/d/1W9W5vSkvMHdwxx6otmp_xH65uyR1u7baMdwlEZEYIvU/edit#gid=0>

Pivotaltracker Link:

<https://www.pivotaltracker.com/n/projects/2487103>

Slack Link:

<https://bumetcs673s21.slack.com/archives/C01LW6FL4SU>

Github Link:

<https://github.com/BUMETCS673/BUMETCS673S21T3>

# Glossary

Heroku: a cloud platform as a service (PaaS) supporting several programming languages. Its function includes deploying, managing, and scaling modern apps.

Agile: a group of software development methodologies based on iterative development.

Scrum: a subset of Agile; a lightweight process framework for agile development.

DevOps: a set of practices that combines software development (Dev) and IT operations (Ops); the combination of practices and tools designed to increase an organization's ability to deliver applications and services faster than traditional software development processes.