**CS673 Software Engineering** 

**Team 3 - ZicZac**

**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 communication and 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**
   1. As a buyer, I want to join in the chat system for sending a message to a seller. (2h)
   2. As a user (buyer/seller) in the chat system, I want to type messages into the message box. (1h)
   3. As a user (buyer/seller) in the chat system, I want to click a send button to send the message. (1h)
   4. As a user (buyer/seller) in the chat system, I want to check the time of sending messages. (1h)
   5. As a user (buyer/seller) in the chat system, I want to receive the message that has been sent from the other user. (1h)
   6. As a user (buyer/seller) in the chat system, I want to see the other users’ username for confirmation. (1h)
   7. As a user (buyer/seller) in the chat system, I want to click a leave button to leave the chat system. (1h)
   8. As a user (buyer/seller) in the home web app, I want to join the chat system by clicking the button (5h).
   9. As a buyer in the chat system, I want to send a direct message to a seller (10h).
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)
5. Log in: as a user, I want to log in to my account, so that I can access my account and orders.
6. Item Categories: As a user, i want to be able to select the category of the items that i want to buy, so that i can filter and see only the group of items i am interested in purchasing.
7. Sort (by price): As a user, I would like to be able to sort items by price so I can choose the cheapest one
8. Customer Service: As a user, I want to be able to submit a complaint about a seller so that i deal with reliable sellers.
9. Post item for sale: As a user, I want to post an item for sale, so that it can be purchased by another user.
10. Review seller: As a user, I want to review a person from whom I bought an item so that others are aware of my experience.
    * 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 Hash

# 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 | Login, signup, homepage with product display & categories | Learning JS, flask integration / project structure  Link category pages to item tags | 125 (~8h per week per person) |  |
| 2 | Functional product page, link buyer & seller via chat system for negotiation/trade | Product page with route to buy, rate, integration of current chat system | 125 |  |
| 3 | Reviews, location based product display, extra features, site deployment | Review function, location tasks, etc | 125 |  |

# 

# Quality Assurance Plan

## Metrics

* + 1. Product metrics: Quality of the product is determined by its reliability, efficiency, number of defects and customer satisfaction. In order to determine the quality of the product, we analyze several product metrics.
* Number of test cases: Unit test, integration test and systems tests will be performed before each alteration. The number of test cases depends on the errors and defects in our software. First test would be performed to find the bugs and then another test would be done to validate the correction.
* Test case pass rate number: This is the percentage of tests that passed successfully
* Number of user story: We have 10 user stories to test so far
* Number of files and classes: So far we have two Python files. One to store user data and one to store inventory. Three classes have been created so far. Account, order and item.
* Customer Satisfaction: Our goal is to have minimum customer complaints 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.

Process metrics:

* Number of defects: Number of defects will be recorded each time the test is done.
* Defect rate: Defect rate will be calculated as number of defects/number of lines.
* Defect fixed rate: This is the number of defects fixed per week.

Project metrics:

* Number of developers:
* Responsibilities: Each software developer is responsible for one of the aspects of the project, however teamwork plays an important role and each developer participates in all aspects of the project. Responsibility assignments, deadlines and productivity plays an important role.
* Team meeting schedule: Once a week, team meeting is conducted to discuss the project. Other times team members communicate through Slack.
* Project goals: User friendliness and security plays an important role in this project. The website has to be attractive to get as many users 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 in place and disqualified sellers who received points below a certain number. In addition, users can contact the customer service in order to complain about a seller.
  + 1. Results:

Iteration 0 - TBD

* 1. Standard

Python flask is mainly used in our project, therefore PEP8 is used as the coding standard.

## Inspection/Review Process

## 

Review will be done mainly by the experienced programmers in the team, however each team member will carefully review and test their code and contribute to system testing. Whenever the code is updated, the team member will create pull request in GitHub and the experienced programmer will carefully review and merge the changes to main branch. Inspection and review will be perform before each alteration. Each member should review every line of code when they update the main branch. The experienced programmer will focus on the lines that have errors. Document review will be done by the QA leader to make sure data is stored and retrieved without errors.

## Testing:

## Link to the testing document: TBD

* Unit testing: Each line of code will be tested by each developer when they update the code. QA leader and experienced coders will also conduct unit testing. Unit testing is the first layer of the testing process.
* Integration testing: All the modules in the program will be combined and tested as a group to make sure that there are no logic errors. Integration testing is the second layer of the testing process
* System testing: System testing is the third layer of the testing process. System testing will be performed to make sure the software is executed without errors and all the functional and non-functional requirements are met.

## Defect Management

* Pivotal Tracker will be used to keep track of the defects.
* QA leader will be mainly responsible for reporting and updating defect status. All team members will contribute to fix the defects.
* Type of defects:

Minor defects: Minor bugs do not affect the overall functionality of the software and can be resolved later. The deadline to resolve minor issues would be before each alteration.

Moderate defects*:* Moderate issues affect the functionality of the software and cause incorrect outputs. Moderate issues should be resolved before minor issues.

Urgent defects: Urgent defects would be the priority. Threat to software or user security and system crashes are examples of urgent issues. Urgent defects should be fixed before any other type of 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:

1. Each username has his/her own branch
2. Each feature from the pivotal tracker has its own branch
3. There is a branch called Development branch that acts as intermediary between any changes ready to commit and main branches.

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
3. Commit was done via pull request first which requires it to be reviewed by peer engineer to ensure it integrates well with the main branch
   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.