**CS673S16 Software Engineering** 

**Team 1 - What’s eat**

**Project Proposal and Planning**

|  |  |  |  |
| --- | --- | --- | --- |
| Team Member | Role(s) | Signature | Date |
| Bowen Jiang | Team Leader/Design Leader | *Bowen Jiang* | 02/07/2018 |
| He Yu | QA Leader | *He Yu* | 02/07/2018 |
| Huaze Sang | Requirement Leader/Implementation Leader | *Huaze Sang* | 02/07/2018 |
| Chengyang Li | Configuration Leader/Environment and Integration Leader | *Chengyang Li* | 02/07/2018 |
| Fengbo Gao | Security Leader | *Fengbo Gao* | 02/07/2018 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Revision history**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Author** | **Date** | **Change** |
| **1** | **He Yu** | **5/2** | **Add contents and graphs** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

[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

Develop and implement a food app that makes daily meal recommendation to our users and also calculate their daily calorie intakes.

# Related Work

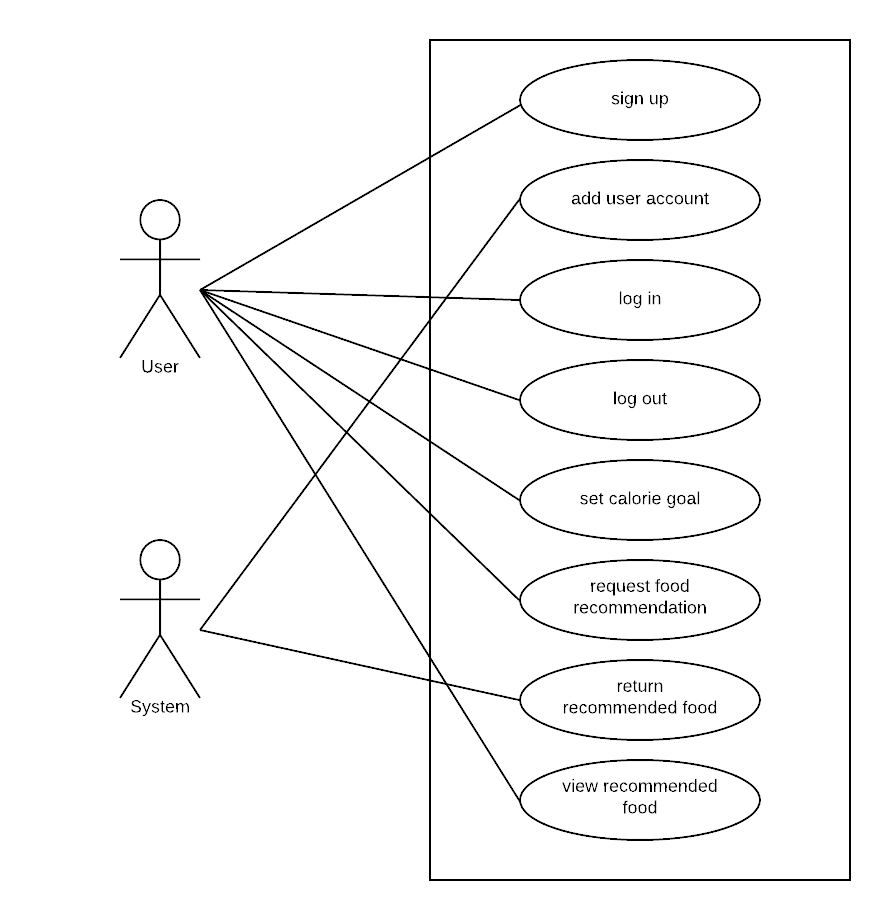
Calories calculator: MyFitnessPal, FitTime

Meal recommendation: <https://www.choosemyplate.gov/>, <http://mofun.c7sky.com/rnd4dinner/>

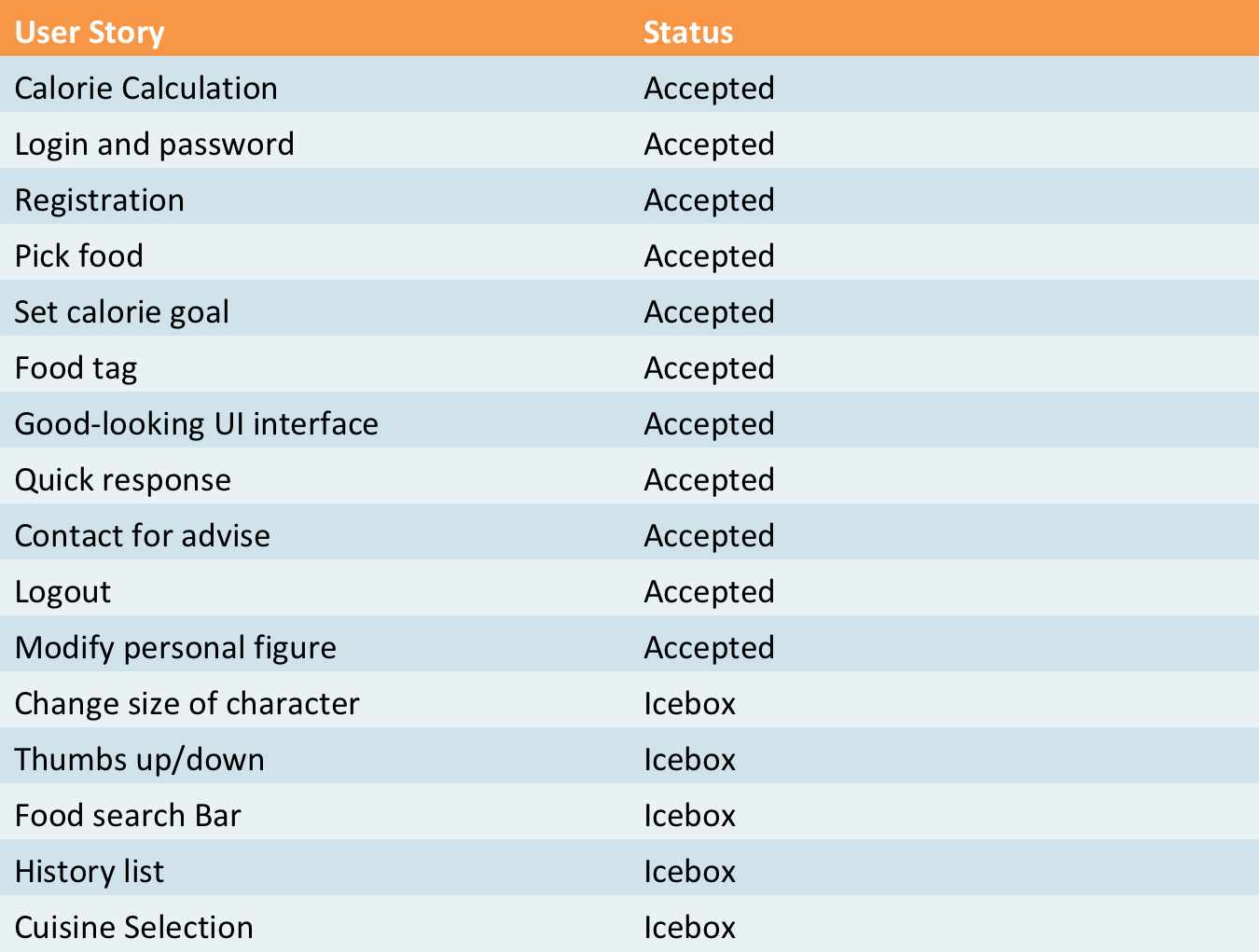
# Proposed High level Requirements

* 1. Functional Requirements
     1. Essential Features (the core features that you definitely need to finish)
* **Website shall have random food selections/ideas just in case our users struggle to pick for themselves.**
* **Website shall allow our users to login and set up daily calorie goal and then show them how many grams they can eat after they made their decisions.**
* **Website shall encrypt users’ private information.** 
  + 1. Desirable Features (the nice features that you really want to have too)
* **Website shall allow our user to set filters when we recommend food to them.**
* **Website shall recommend food to our users based on their past decisions instead of randomly picking up.**
  + 1. Optional Features (additional cool features that you want to have if there is time)
* **Take other nutrients e.g. protein and vitamin, into consideration when recommending food to our users.**
  + 1. Existing Features (Not applied to a brand new project)

The graph below shows the use case of our project:



* 1. Nonfunctional Requirements
* **Quick response**
* **Good UI**
* **Easy to access and register**
  1. Implemented Features (*to be completed at the end of each iteration*)



# Management Plan

# (For more detail, please refer to SPMP document for encounter example)

## Process Model

**Agile-Scrum**

1. **Documents and artifacts:**

**Pivotal tracker:** [**https://www.pivotaltracker.com/n/projects/2149232**](https://www.pivotaltracker.com/n/projects/2149232)

**Product backlog;**

**Sprint backlog;**

**Burndown chart;**

**The burndown chart is get form pivotal tracker:**

1. **Roles:**

**Product owner(All memebers);**

**Scrum master: Bowen Jiang;**

**Developer(all team members);**

**Team leader: Bowen Jiang**

**Tech leader: Huaze Sang;**

**QA/Document leader: He Yu;**

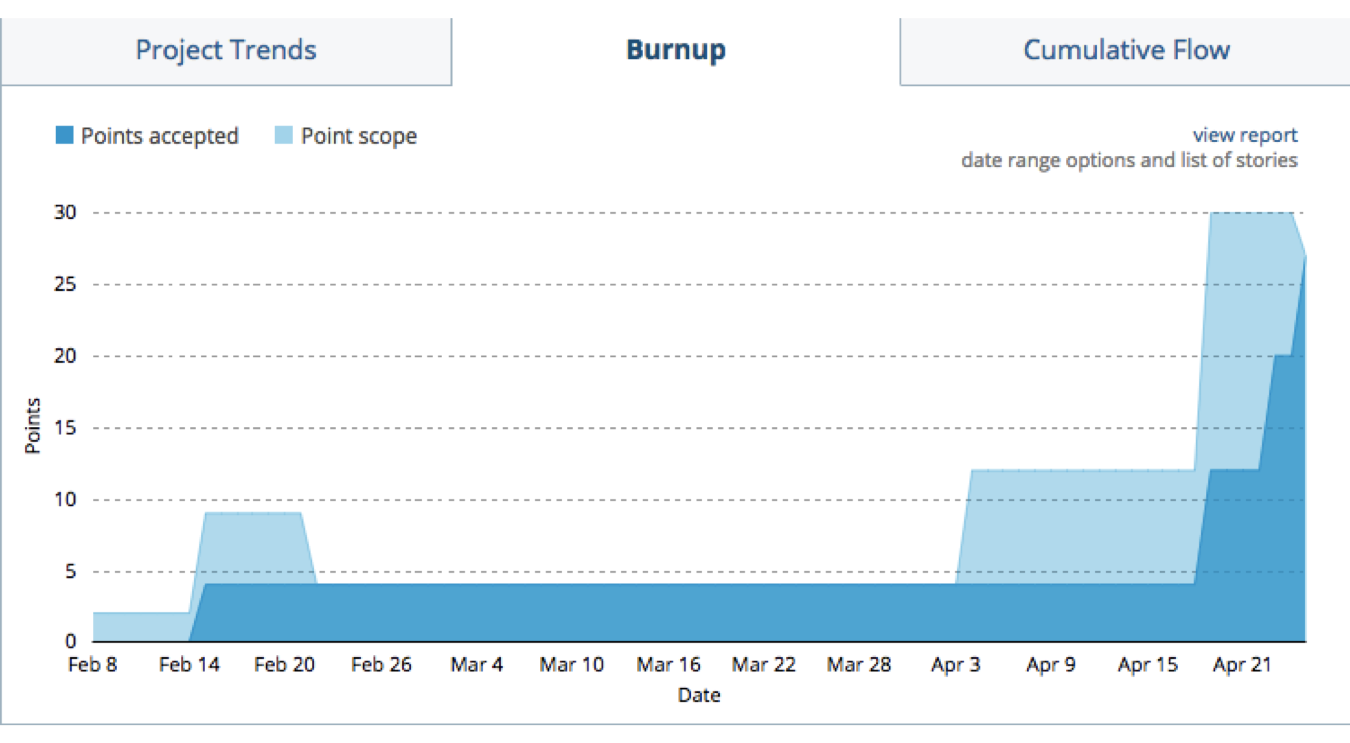
**Front-end design leader: Chenyang Li;**

**Security leader: Fengbo Gao;**

1. **Process:**

**All the processes are recorded by pivotal tracker, the following graph shows the project trends and burnup chart (instead of burndown chart):**

****

****

## Objectives and Priorities

(Project Goals can include but not limited to completed all proposed (essential) features, deploy the software successfully, the software has no known bugs, maintain high quality? )

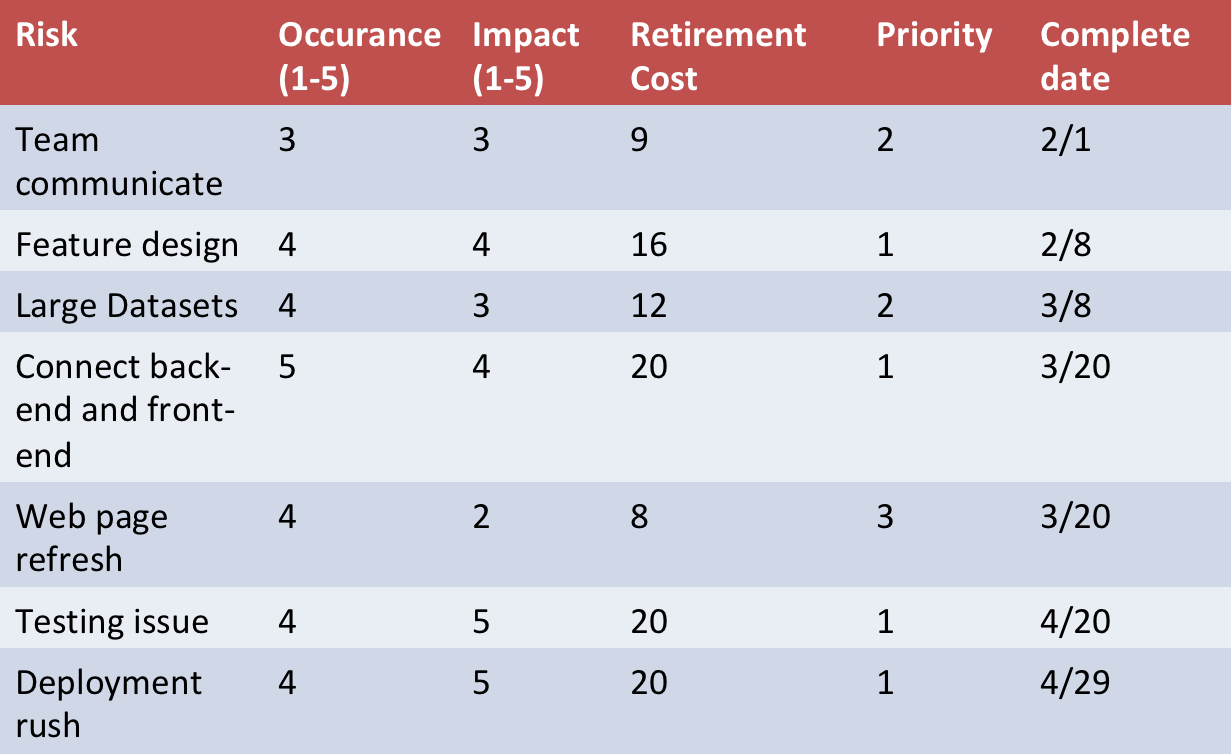
**1- maintain high quality**

**2- complete all essential features**

**3- deploy software successfully**

**(no known bugs -> quality metrics )**

## Risk Management (need update constantly)



## Monitoring and Controlling Mechanism

**- Use github and pivotal tracker to keep track of the process;**

**Pivotal tracker:** [**https://www.pivotaltracker.com/n/projects/2149232**](https://www.pivotaltracker.com/n/projects/2149232)

**Github: https://github.com/BU-CS673-S18T2/whats-eat-app**

**- Individual weekly work report;**

**- Weekly face-to- face meeting(meeting minute document):**

**Weekly process and status reports;**

**Update risk management document weekly;**

**- Quick(daily) status report and team feedback through slack and wechat;**

**- Burndown chart;**

**- Test and maintenance report for each requirement;**

## Schedule and deadlines (need update constantly)

**Iteration 0: 1/18 - 2/8 Inception - Planning phase**

**Iteration 1: 2/9 - 3/2 Elaboration- Design and implement**

**Iteration 2: 3/3 - 3/30 Construction - Implement and test**

**Iteration 3: 3/31 - 4/27 Transition - Integration and test**

**(In the end of every iteration, there should be sprint-review/ task assignment for next iteration).**

# Quality Assurance Plan

# (For more detail, please refer to SQAP document for encounter example)

## Metrics

* + 1. Definition

1. Product Metrics

* Test coverage rates
* Total lines of code
* Number of classes
* Number of bugs

1. Process Metrics

* Working hours per man per week
* Estimated deadline and actual completion time
  + 1. Results (to be completed at the end of each iteration)

## Standard (e.g. documentation standard, coding standard etc. )

## Inspection/Review Process

## (e.g. describe what are subject to review, when to conduct review, who do the reviews and how ?)

***The subjects to review*: check through the product/whether it works as we expect. A discussion about the current issues we have and make a possible solution during the meeting. Then discuss our solution with the professor and get some advice.**

***When*: Each week, we have at least 2-hour meeting before the class. And if we have extra work such as presentation in class we will have an extra 2-hour meeting. Each time we have the meeting, we will report the** **problems for each person. Sometimes members in the team will have different opinions for one problem.**

***Who*: All the team members should discuss the problem and maybe ask professor for help if we have problems cannot solved during the meeting.**

**We also need to focus on revising the current work we have. Each group member’s work should be discussed during the meeting and get advice from other. Some changes should be added to the work after the meeting.**

## Testing

***Who*: All the team members have the responsibility to test the work of the project. Each member should not only test his own work but also test work of other group members and give some suggestions to revise.**

***When***: **During the weekly 2-hour meeting, we discuss the test of the work of each member for last week. Especially when we have every iteration done or revise our software.**

***Types***:

**According to the website:****<https://www.codeproject.com/Tips/351122/What-is-software-testing-What-are-the-different-ty>**

***Unit testing*: testing of an individual unit or group of related units.**

***Integration testing*: testing in which a group of components are combined to produce output and the interaction between software and hardware.**

***Functional testing*: testing to ensure that the specified functionality required in the system requirements works.**

***System testing*: testing to ensure that by putting the sofrware in different environments it still works.**

***Performance testing*: testing to assess the speed and effectiveness of the system and to make sure it is generating results within a specified time as in performance requirements.**

***How*: We will use a testing tool to help us with each testing above. We will have one person to records all the testing result and save them in a document. Each meeting we will update our testing and testing records document.**

## Defect Management

(e.g. describe the criteria of defect, also in terms of severity, extend, priority, etc. The tool used to management defect, actions or personnel for defect management)

**After do the testing, we must have some defects or bugs in our work.**

**The severity of a defect is related to how severe a bug is.**

**The priority is a defect is related to how quickly a bug should be fixed and deployed to live servers.**

**In our team, we determine the priority of the bugs depending on the severity of the bugs. Usually, the higher severity the bug have, the higher priority the bugs have. We have to focus on the high severity bugs so that we may have our software works and then we will deal with the low severity bugs so that our software works correctly as we expect.**

**The tools we use:**

***Reporting the bugs*: After the testing, the team will focus on the debugging. Every group members finds a bug, he has to report it and record it before fixing it.**

***Assigning works*: After we see the bugs in the reporting, we will assign the debugging work to specific team member.**

***History/Work comments*.**

***Graphs/tables***

***Contacting*: each team member should contact with each other when he does his debugging work. The team can also list some bugs and contact with professor for advice.**

# Configuration Management Plan

(For more detail, please refer to SCMP document for encounter example)

## Configuration items and tools

* + 1. **Version control tools: git  
       Project link:** [**https://github.com/BU-CS673-S18T2/whats-eat-app**](https://github.com/BU-CS673-S18T2/whats-eat-app)
    2. **Issue (bug) control tool: Github issues**
    3. **Requirement analysis and progress monitor tools: Pivotal tracker  
       Project link:** [**https://www.pivotaltracker.com/n/projects/2149232**](https://www.pivotaltracker.com/n/projects/2149232)
    4. **Deployment: To be determined.**
    5. **Communication tools: Wechat.**
    6. **IDE: Eclipse.**

## Change management and branch management

* + 1. **All developers should have their own branches**
    2. **All developers should develop on their own branches**
    3. **A ‘‘master’’ branch is created to hold published/functional version of project.**
    4. **Master branch should only be merged by the assigned Design Leader.**

## Code commit guidelines

* + 1. **Rebase from master branch before everyday’s work**
    2. **Commit constantly to local**
    3. **Push to developer’s own branch to wrap up daily’s work**
    4. **File merge request to configuration leader every time a merge to master branch is needed**
    5. **Configuration leader will perform merging as needed**

# References

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

# Glossary