PROJECT MANAGEMENT PLAN

FOR THE

PERSONAL FOOD LOG APP

VERSION 1.0.0

NOV 1ST 2019

ELG 5100 Group 5

Yuhao Shen 300086198
Yi Pang 101100793
Xu Zhang 300062651
Ke Yang 300087074

RECORD OF CHANGES

*A - ADDED M - MODIFIED D – DELETED

VERSION NUMBER	DATE	NUMBER OF FIGURE, TABLE OR PARAGRAPH	A* M D	TITLE OR BRIEF DESCRIPTION	CHANGE REQUEST NUMBER
1.0.0	Nov 1st 2019	/	A	The first version	/

TABLE OF CONTENTS

SECTION 1. OVERVIEW

1.1PROJECT SUMMARY

1.1.1 Purpose, Scope, and Objectives

Purpose:

Build a mobile app that can detect the food and calculate calories to estimate intakes for the user of DH inc.

Objectives (software):

1. The project software can realize the function of ingredient detection
2. The project software can realize the function of calories calculation
3. The project if for a mobile App publication

Scope:

1. The project is responsible for the delivering of a mobile application for IOS or Android based systems.

2. The project software for user works on the following functions:

1). Food logging, taking food pictures and analyze it then stored in database.

2). Ingredient identification, using deep learning technique to detect and identify food ingredients.

3). Calories calculation, calculating the calories get from the analyzing function.

3. The software product works associated with the techniques assigned by Digital Health.Inc.

1). Smartphone based VBM system. Where the users simply take a picture of the food, and the app calculate the number of calories and nutrition in that food.

2). During the development procedure, a very promising system with accuracy of 94.11% will be built in the software development.

3). Our project management team will be reunited in product department of Digital Health.Inc. With its 19 software development professionals.

1.1.2 Assumptions and Constraints

No.	Assumptions
1	It is assumed that all the relevant licenses and development softwares are allowed to use, including Axure RP 9, React Native, AWS Amplify Jupyter Notebook, Selenium, Git, Workfront online tool, Microsoft Office, Google Docs and Google Drive, SmartSheet and Microsoft Project.
2	It is assumed that all the required hardware are available , test Android operating system smartphones and IOS operating system smartphone, computers
3	It is assumed that the office or a lab is available.
4	It is assumed that the developer master all the programming language we use in this project.

No.	Constraints	
1	Requirements	Client provide all the requirements
2	Techniques	Development people comes from the Health.ca Inc
3	Schedule	Fixed deadline for the Project Plan
4	Private Policy	Compulsive policy to obey
5	Developer's abilities	all the development process are done by the assigned develop team of Digital Health Inc

1.1.3 Project Deliverables

Project Deliverable 1: Final mobile App	
Stakeholder:	Digital Health Inc
Description:	Final mobile application is a software works on mobile devices allowing users to know calories in food involved.
Acceptance Criteria:	The final mobile must satisfy all the relevant requirements set Digital Health Inc.(released in Android and IOS app store)
Project Deliverable 2: User's manual	

Stakeholder:	Digital Health Inc
Description:	A user instruction to explicit the functions of the mobile application and how to use it
Acceptance Criteria:	The user's manual must be documented and clearly cover all description of the mobile application(in book)
Project Deliverable 3: Host server login credentials	
Stakeholder:	Digital Health Inc.
Description:	It is about configuration setup on host server, including account on AWS and relevant documents.
Acceptance Criteria:	AWS accounts must be delivered with the correct username and password, and the documents must explicit the relevant works clearly.(in DVD)
Project Deliverable 4: Maintenance guide	
Stakeholder:	Digital Health Inc.
Description:	A document about the bugs and issues encountered in project software development will be delivered to Digital Health Inc at the end of the project.
Acceptance Criteria:	Bugs and issues should be documented clearly with related solutions if applicable.(in book)
Project Deliverable 5: Source code	
Stakeholder:	Digital Health Inc

Description:	All the relevant source code of the mobile application will be delivered at the end of the project.
Acceptance Criteria:	Full-functional source codes are required(in DVD)

1.1.4 Master Schedule and Budget Summary

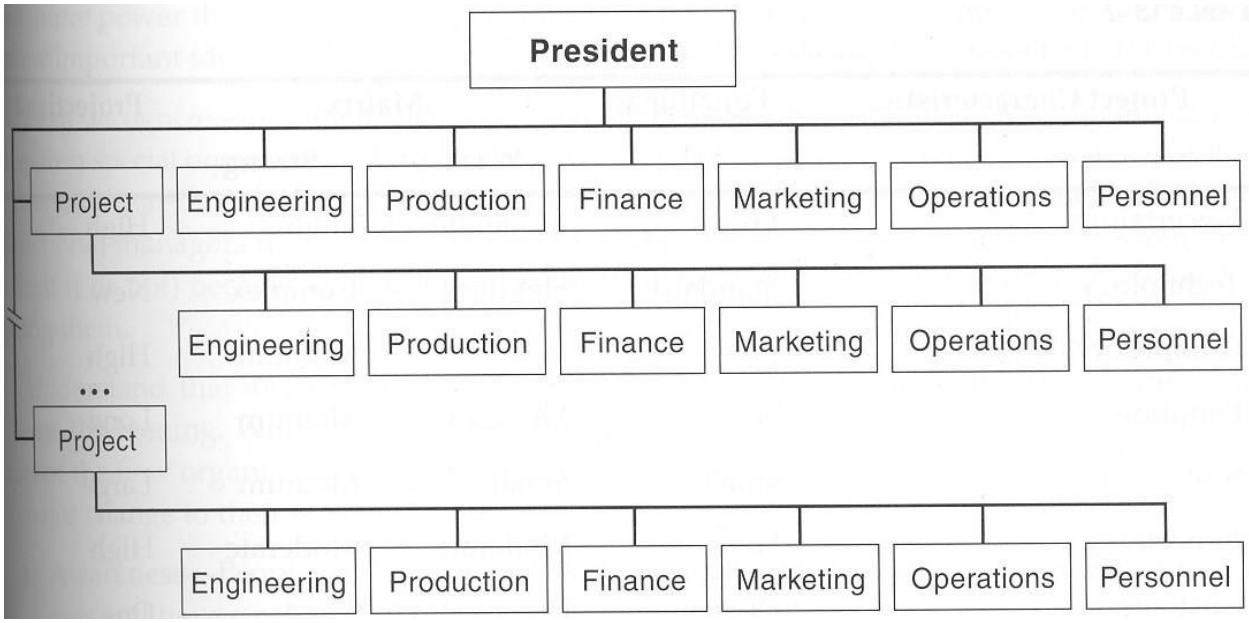
Major work activities And Supporting processes		**Summary of schedule**	**Summary of budget (CAD)**
Pre-work	Personal preparation	5 weeks Nov 3-Dec 6	5,000
	Development preparation		
Development	Ingredient detection	30 weeks Dec 8-Jul 5	1,838,802
	Convert volume from weight to calories		
	AWS back-end		

	React native front-end		
	Release test		
	Get volume		
Delivery	Documents	5 days Jul 7-Jul 12	N/A
	Codes		

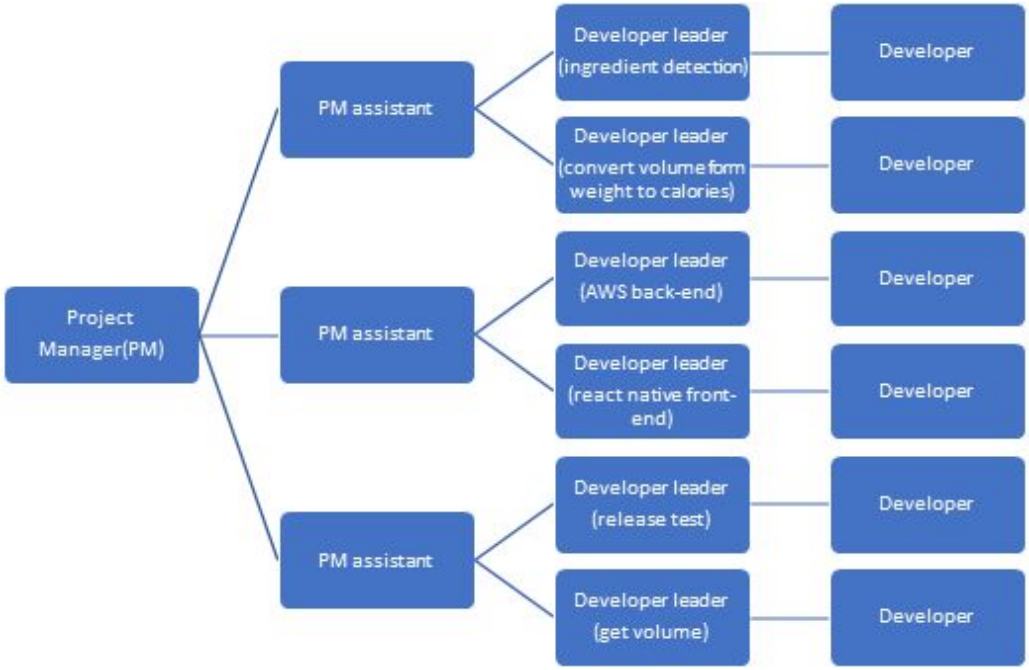
SECTION 2. PROJECT ORGANIZATION

2.1EXTERNAL INTERFACES

Since we are using agile methodology to conduct our project. We use projectized structure as external organization that all skills needed are assigned to the project full-time, and sill specialist report to project manager only.



2.2INTERNAL STRUCTURE



2.2.1 The Project Manager

2.2.1.1 Scope of Authority.

The project manager's role is to ensure all tasks have been assigned and proceed as planned.

2.2.1.2 Scope of Responsibility.

The project manager needs to be responsible to follow the progress, keep the schedule, coordinate groups and control budget.

2.2.1.3 Internal Responsibilities.

It is the project manager's responsibility to arrange working plan, project controls. Project manager also needs to conduct risk management as well as configuration management.

2.2.1.4 External Responsibilities.

It is the project manager's responsibility to coordinate to coordinate a diverse group of people to work with management, engineering, accounting, financing, public relations, construction, purchasing, design, and all other groups.

2.3PROJECT ROLES AND RESPONSIBILITIES

Roles	Responsibilities
Project Manager	Software project planning and monitoring, Milestone and schedule planning and monitoring, Set and communicate the team meeting agendas. And risk Management.
Project Assistant	Help project manager to gather documents and be responsible for monitoring the assigned develop tasks
Developer Leader	It is the developer leader's responsibility to control the developing schedule and coordinate the developing problems directly, developer leader also needs to guarantee the completeness of the assigned features.
Developer	Developers should be responsible for the completeness of the tasks assigned by the developer leader.

SECTION 3. MANAGEMENT PROCESS

3.1START-UP

3.1.1 Estimation

The basis of estimation we use for this project is analogy and local history since we have

several professional developers and analysts with previous similar experiences. To get the estimation of the cost and schedule, we will have the estimation of project size first. We applied Beta Distribution Measurement [Line of Code = (Optimistic + Pessimistic + 4 * Realistic) / 6] to get estimation for our project, since this is a relatively confident estimation. Initially, we got 6 main features in the project, including back-end, front-end, testing, etc. Back-end code is on the pessimistic level since it's the most crucial part of our project. Front-end code is on the optimistic side, and the testing is on the realistic side. Finally, the line of code for this project is about 16k line of code.

According to the project size, we will have a schedule estimation. We will do this project in the Scrum model. Before the actual working, we will need time to recruit for subcontract designer and testing staff, resources acquisition, also training for all developers, designers, assistants, and testing staff. The whole implementation phase of the project is divided into several features, so each feature can start simultaneously. We will have a release every 2 weeks. For every 2 weeks, 5 days are for really developing to release, 3 days for testing, 1 day for verification with our customer, 1 last day for planning for the successive week. In this manner, our schedule estimation for the whole project is 34 weeks (around 8 months). The confidence level for the schedule is medium. If there's no feature change from customer in the future or serious risks, this estimation would be reliable.

Resource requirement estimation would mainly be personnel for the project. Hardware, software, licenses, contracts, facilities, etc. are achieved as an assumption. We divide our personnel team according to the features. Finally, we have 19 developers from Digital Health Inc. for all back-end features and AWS back-end development, 6 UI designer in charge of design and front-end development from the subcontractor, 6 test staff in charge of test before each release and test for final release from the subcontractor, 1 project manager, and 3 assistant. The confidence level is medium since it's dependent on project size.

With an appropriate estimation of schedule, our main budget consumption will be the salary for our team, some activity fees, and support fees. All software, hardware, licenses, contracts are assumed already equipped in assumption since Digital Health Inc. has already developed some similar mobile App. The salaries of each developer, designer, test staff, project manager, etc. are calculated according to the average salary of each position in the Ottawa area. The budget for salaries and other activities fees and the supporting fees would be approximately $1,900,000 CAD. The confidence level for the budget is pretty high since the size estimation and staff estimation are quite reliable.

3.1.2 Staffing

- AI (Artificial Intelligence) staff (for food detection)
    - Proficient in python, and CNN, 2 programmers with experience and 2 programmers with less experience in the CONSTRUCTION phase. 1 analyst with extensive experience.
- Dimension estimation – get volume
    - Proficient in deep learning, mobile cloud computing, distance estimation and size

calibrations inside mobile devices, 1 programmer with experience and 1 programmer with less experience in CONSTRUCTION phase.

- Weight calculation by volume and convert to calories
    - 1 programmer with experience and 1 programmer with less experience in python language and algorithms in the CONSTRUCTION phase.
- Back-end Developer
    - ports connection
        - Proficient in AWS back-end development. 2 programmers with experience and 2 programmers with less experience in CONSTRUCTION phase + 1 programmer who has extensive experience
    - database control
        - Big data and cloud computing, familiar use of Amazon Cloud and deep learning, 1 programmer with experience and 1 programmer with less experience in CONSTRUCTION and TRANSITION phase
- Support Developer
    - 1 programmer with experience in all phases.
- UI designer front-end
    - Proficient in React Native front-end, PS, 6 designers in CONSTRUCTION/ELABORATION phase (sub-contract)
- Manager Assistant staff (PM team)
    - Project Manager: proficient in project management skills and tools.
    - 3 assistants: proficient with Microsoft Word, Excel, and PowerPoint. Good at communication and solving conflicts, and working well with others.
- Testing staff
    - Proficient in Automatic test , 3 programmers with experience and 3 programmers with less experience in INCEPTION and TRANSITION phase (sub-contract).

3.1.3 Resource Acquisition

- Personnel

Gathering all personnel will start right after the project plan finish, Nov. 4, 2019, finish the subcontract with UI designers and test staff on Nov. 18, 2019. The ability of UI designers and test staff may not be satisfying. The contract will end if they could not pass the evaluation in the future phases.

- Manager + 3 assistants: 3 assistants will monitor 2 features independently for document collect and meeting minutes.
- Developers for AWS back-end: we will mainly use the developers offered by Digital Health Inc. Each of them will be assigned to different feature teams as described in the previous Staffing sub-clause.
- For UI designers: we will use professional subcontractor business to hire 6 designers in charge of the UI design and React Native front-end development. The hiring process will be done before the start of the training process.

    - For the Testing team: we will use professional subcontractor business to hire 6 testing staff for each iteration of integration testing and final testing. The hiring process will be done before the start of the training process.
- Equipment, hardware and software

All equipment, hardware and software mentioned below are assumed already equipped in the assumption, so there's no acquisition plan for these.

    - Each developer will be equipped with a computer, installed with python and functional AWS back-end environment.
    - Each UI designer will be equipped with a computer, installed with PS and functional React Native environment.
    - Project Manager and 3 assistants will be equipped with a PC, installed with Microsoft Word, Excel, PowerPoint.
    - Auto testing will be done on an online website tool, Selenium.
    - Each test staff will be equipped with an android phone for the release test.
- Service and Contracts
    - AWS service, license achieved in assumption.
    - Workfront -- our project, budget, schedule tracking tool, license achieved in assumption.
    - Smartsheet -- our schedule tool, license achieved in assumption.
    - UI designer team, we will contact a profession subcontractor team to get a professional UI design team.
    - The test staff team, we will contact a profession subcontractor team to get a professional test team.
- Facilities
    - Office building, others are mentioned in assumption.
- Administrative
    - AWS back-end database is only available to developers and project manager.
    - The algorithm and source code is only available to certain feature teams.
- Janitorial services
    - Smartsheet, the progress tracking tool is only available to the project manager.
    - developers of Digital Health Inc. will take in charge of future maintenance.

3.1.4 Staff Training

- Training for developers (3 weeks)

Since we apply the Scrum model, so we have Pair programming, so we will have the experienced developers as mentors to guide other developers to this training and finally working well with each other.

    - For AI and deep learning
        - Method of training: mentoring (with experience developers) and computer-assisted training.
        - Number of personnel: 4

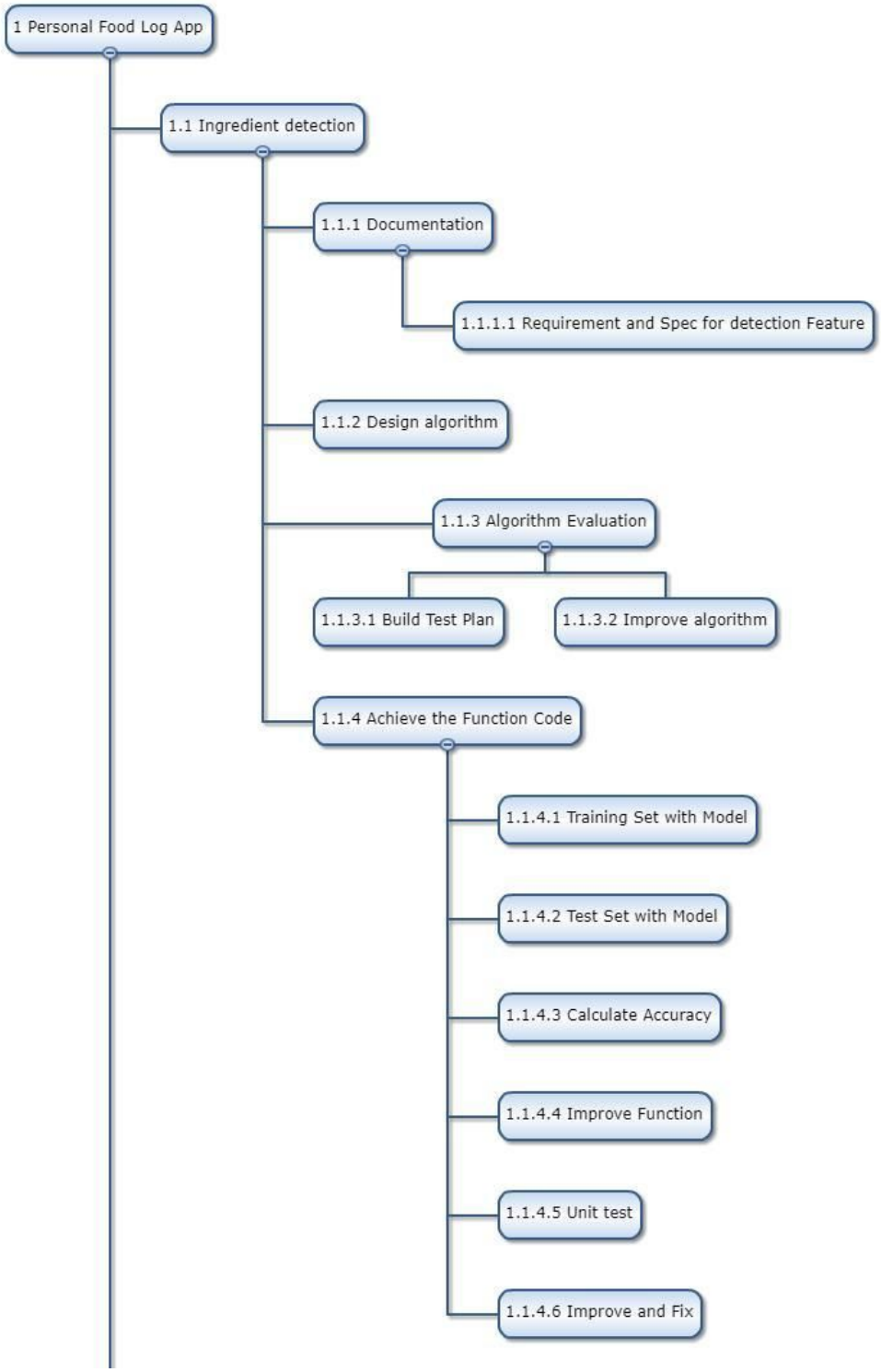
        - Entry-level: has previous experience with python developing, know the basic idea of deep learning.
        - Exit level: familiar implementation of deep learning with python, able to work well with mentors.
    - For volume calculation
        - Method of training: mentoring (with experience developers) and computer-assisted training.
        - Number of personnel: 4
        - Entry-level: knowledge about deep learning and mobile cloud computing.
        - Exit level: know distance estimation and size calibration inside an Android mobile device, able to work well with mentors.
    - For convert volume to calories
        - Method of training: mentoring (with experience developers) and computer-assisted training.
        - Number of personnel: 2
        - Entry-level: has the previous developing experience, coding experience
        - Exit level: know the basic working flow of AWS, able to work well with mentors.
    - For big data and cloud computing
        - Method of training: mentoring (with experience developers) and computer-assisted training.
        - Number of personnel: 2
        - Entry-level: has previous experience with Amazon Cloud or know the basic idea of deep learning cloud computing.
        - Exit level: familiar use of Amazon Cloud, able to work well with mentors.
    - For AWS back-end developing
        - Method of training: mentoring (with experience developers) and computer-assisted training.
        - Number of personnel: 2
        - Entry-level: has the previous developing experience, coding experience
        - Exit level: know the basic working flow of AWS and know how to code for mobile App.
- Training for UI designer (1 week)
    - Method of training: lecturing and computer-assisted training
    - Number of personnel: 6
    - Entry-level: has previous experience of mobile UI design, have a basic knowledge of React Native front-end frame.
    - Exit level: get the idea and requirements of our App, know the reporting schedule and format for our project.
- Training for testing staff (1 week)
    - Method of training: computer-assisted training
    - Number of personnel: 6
    - Entry-level: has previous test experience for mobile App.
    - Exit level: know how to use the online Selenium tool, know the reporting

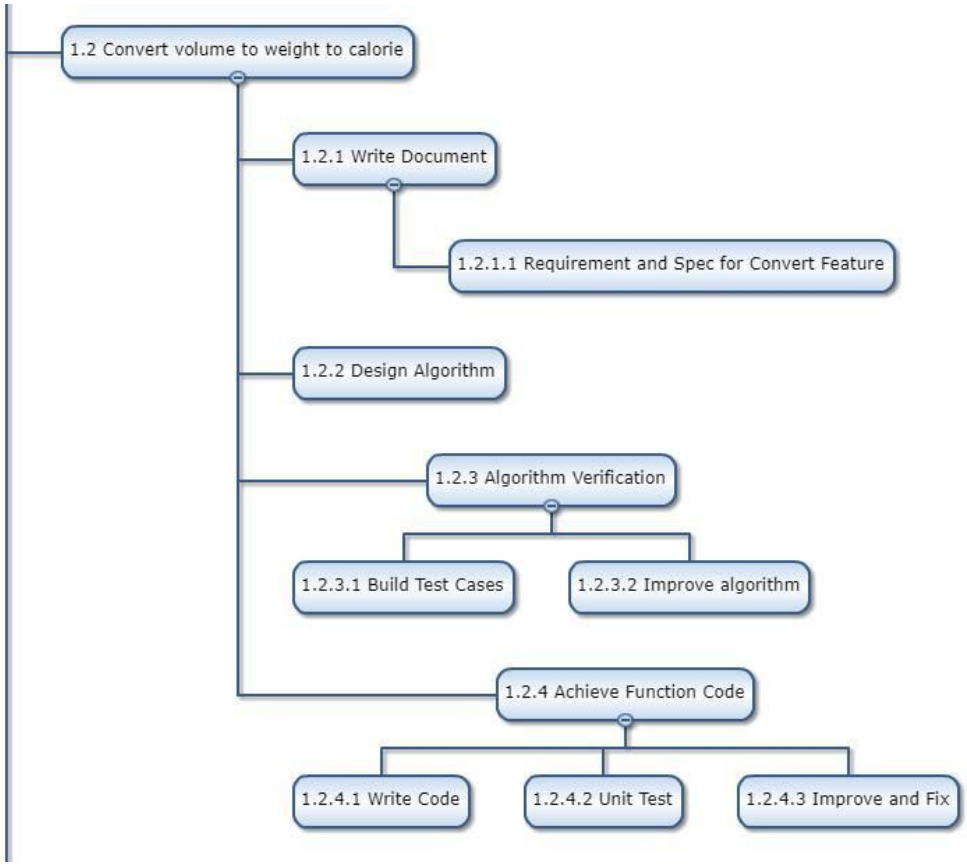
schedule and format for our project.

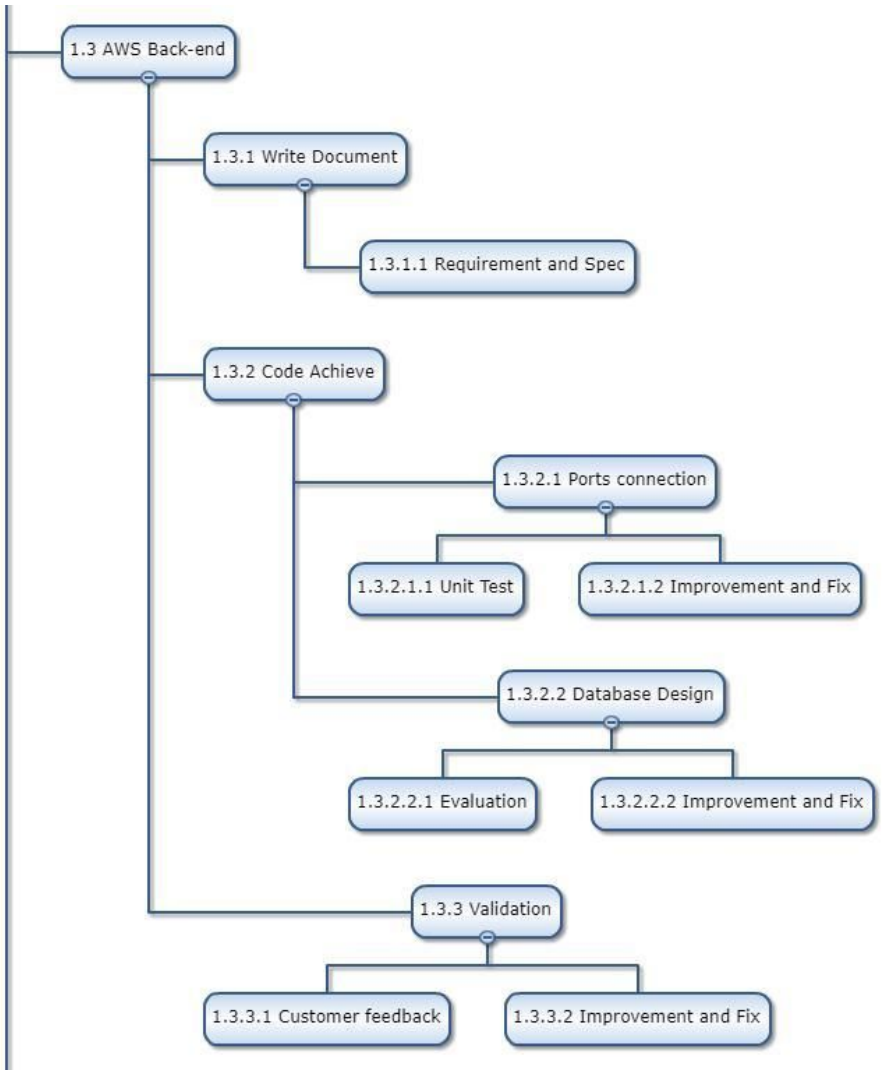
- Training for assistants (1 week)
    - Method of training: lecturing
    - Number of personnel: 3
    - Entry-level: has previous experience working for a project
    - Exit level: know the document management style, meeting minutes style.

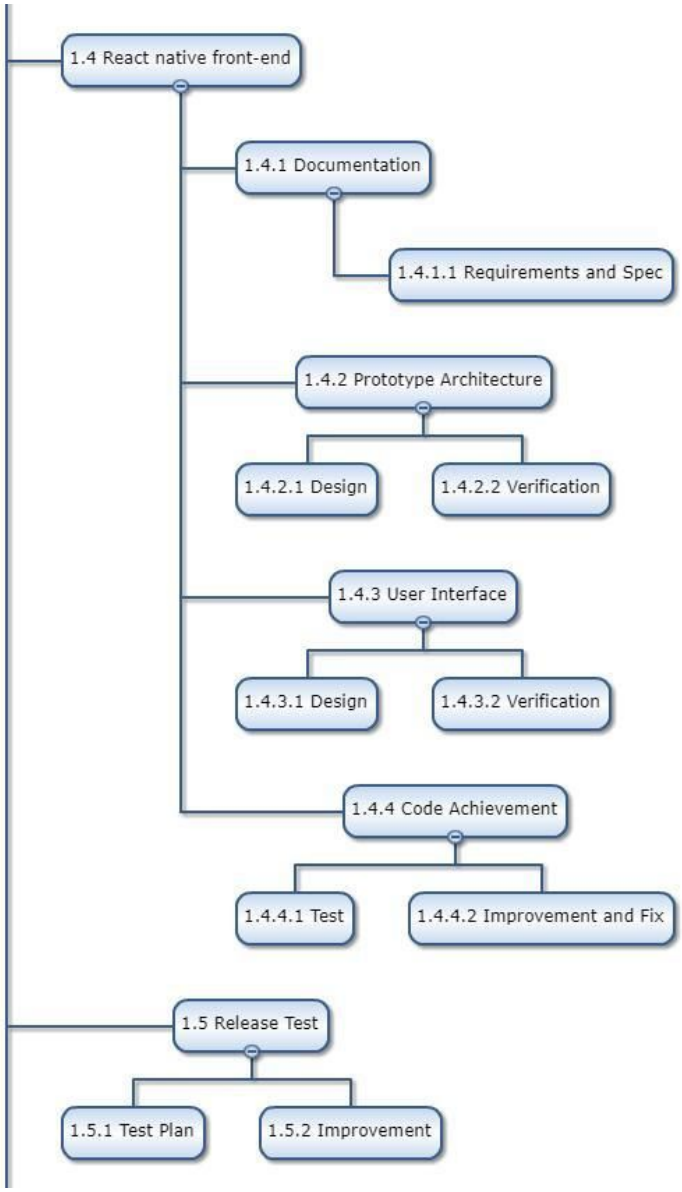
3.2WORK PLANNING

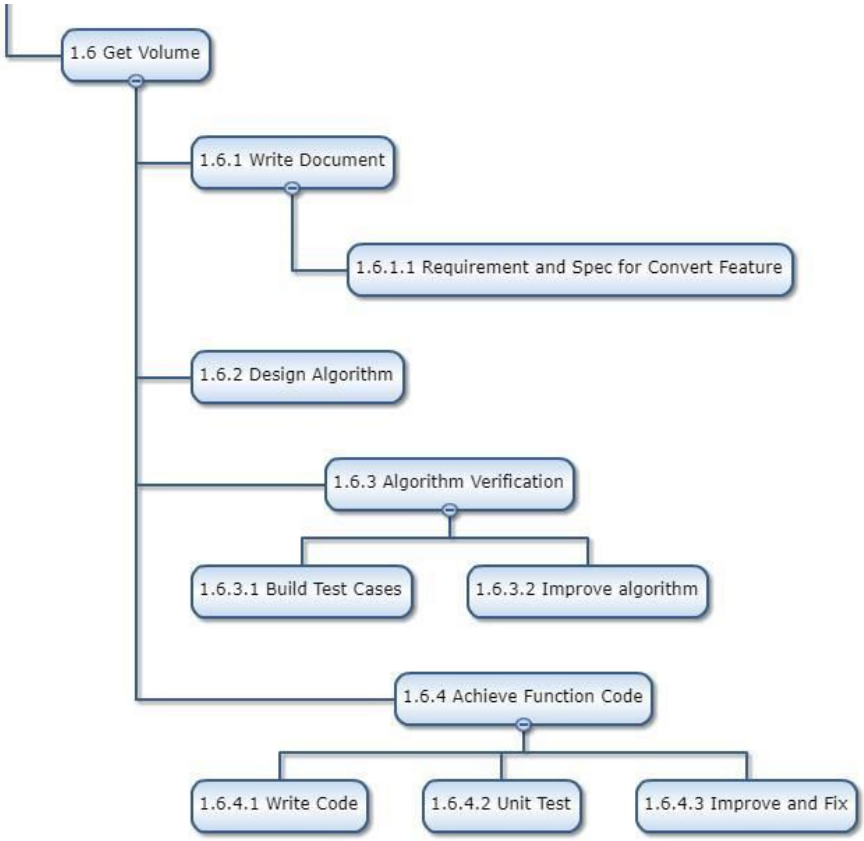
3.2.1 Work Activities


1 Personal Food Log App
1.1 Ingredient detection
1.1.1 Documentation
1.1.1.1 Requirement and Spec for detection Feature
1.1.2 Design algorithm
1.1.3 Algorithm Evaluation
1.1.3.1 Build Test Plan
1.1.3.2 Improve algorithm
1.1.4 Achieve the Function Code
1.1.4.1 Training Set with Model
1.1.4.2 Test Set with Model
1.1.4.3 Calculate Accuracy
1.1.4.4 Improve Function
1.1.4.5 Unit test
1.1.4.6 Improve and Fix


1.2 Convert volume to weight to calorie
1.2.1 Write Document
1.2.1.1 Requirement and Spec for Convert Feature
1.2.2 Design Algorithm
1.2.3 Algorithm Verification
1.2.3.1 Build Test Cases
1.2.3.2 Improve algorithm
1.2.4 Achieve Function Code
1.2.4.1 Write Code
1.2.4.2 Unit Test
1.2.4.3 Improve and Fix

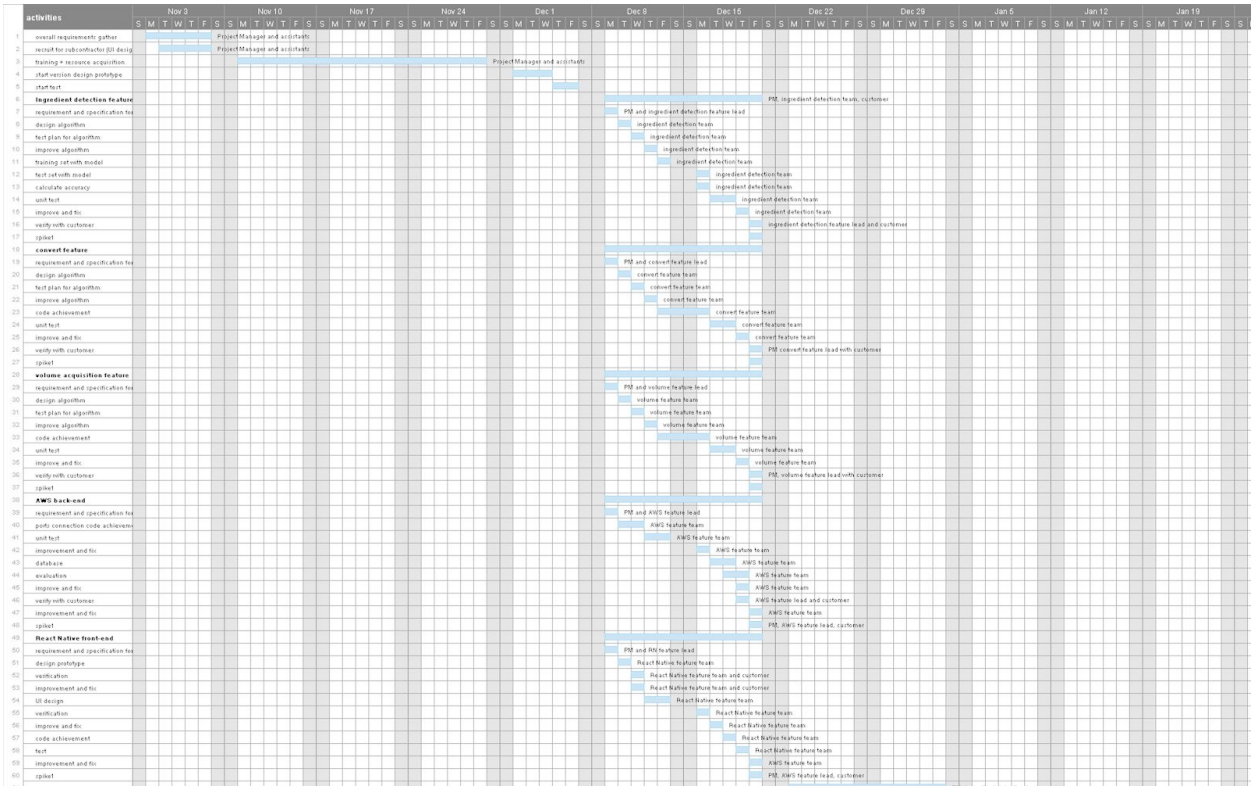

1.3 AWS Back-end
1.3.1 Write Document
1.3.1.1 Requirement and Spec
1.3.2 Code Achieve
1.3.2.1 Ports connection
1.3.2.1.1 Unit Test
1.3.2.1.2 Improvement and Fix
1.3.2.2 Database Design
1.3.2.2.1 Evaluation
1.3.2.2.2 Improvement and Fix
1.3.3 Validation
1.3.3.1 Customer feedback
1.3.3.2 Improvement and Fix


1.4 React native front-end
1.4.1 Documentation
1.4.1.1 Requirements and Spec
1.4.2 Prototype Architecture
1.4.2.1 Design
1.4.2.2 Verification
1.4.3 User Interface
1.4.3.1 Design
1.4.3.2 Verification
1.4.4 Code Achievement
1.4.4.1 Test
1.4.4.2 Improvement and Fix
1.5 Release Test
1.5.1 Test Plan
1.5.2 Improvement



3.2.2 Schedule Allocation

We are planning our schedule with the Scrum model, so the whole process is like a cycle. We divide it into 14 releases for development. Before all that, there's time for resource gathering and training. After all 14 releases, there will be a release test and then the final release. The following image shows the schedule for 1 release, the whole detailed schedule is in the attached 5100.xml file. Some national vacations are not indicated in this schedule. The process will automatically postpone for those days.



3.2.3 Resource Allocation

Features	number of personnel	skill level	hardware	service	facilities
start version prototype	6 UI designers	intermediate	computer	PS, InVision	building, office supplies
start test	6 test staff	intermediate	computer	selenium	building, office supplies
ingredient detection	PM + 1 assistant + 4 developers + 1 expert	2 experienced + 2 less experienced	computer, smart phones	AWS, selenium	building, office supplies
dimension estimation -- get volume	PM + 1 assistant + 2 developers	1 experienced + 1 less experienced	computer, smart phones	AWS, selenium	building, office supplies
convert volume to calories	PM + 1 assistant + 2 developers	1 experienced + 1 less experienced	computer, smart phones	AWS, selenium	building, office supplies
AWS back-end	PM + 1 assistant + 6 developers +1 extensive	3 experienced + 3 less experienced	computer, smart phones	AWS, selenium	building, office supplies
React Native front end	PM + 6 UI designers + 1 assistant	intermediate	computer, smart phones	PS, InVisionReact Native, selenium	building, office supplies
release test	PM + 6 test staff + 1 assistant	intermediate	computer, smart phones	Selenium	building, office supplies

3.2.4 Budget Allocation

	position	skill level	number of personnel	sub-total (CAD)
salaries	Project Manager	senior	1	63,333
	assistants	intermediate	3	112,000
	back-end developers	experienced	8	469,973
	back-end developers	less experienced	7	364,000
	UI designer	intermediate	6	292,000
	test staff	intermediate	6	324,000
	developer analyst	senior	2	114,666
	software experts	senior	1	90,000
activities	once a month, 200 per time			1,600
holiday gifts	3 times within the whole project			3,150
work overtime	maximum 5 hours per week for the whole project team			4080
total				1,838,802

3.3PROJECT CONTROLS

3.3.1 Requirements Control

We applied the Scrum model to complete our project, so we have a really close connection with our customer. We know that change is inevitable within the process, so we set up multiple meetings with our customer to ensure the right design and developing process.

Approximately, we planned 14 releases to finish the whole project. There will be a meeting with the customer every two weeks to verify the current developing process and new ideas. When we get something new, firstly, we will go through the project plan to check whether the new requirement is within the scope. If it's without the scope, there would be a re-estimation of schedule, budget, and plan to deal with the new features. If we can get a positive response from the customer with the new estimation, we will have a formal report document, notify the corresponsive team, and make revisions according to the estimation. The impact on risk might happen because of the ability of the subcontract.

We will develop prototype and algorithm models for each requirement change as verify tool with the customer.

3.3.2 Schedule Control

3.3.2.1 Schedule Tracking.

We will use Smartsheet as our schedule tracking tool to track the percentage of complete and status of each work activity. The completion of each work activity in every release is the pass of both verification and validation test.

3.3.2.2 Schedule Performance Reports.

There will be a performance report every two weeks about the working process from every developer, UI designer, and test staff.

3.3.2.3 Schedule Reviews.

A formal meeting every two weeks talks about the current progress according to the collected report from the working staff. Each feature lead and project manager will adjust the future schedule according to the current working progress.

3.3.2.4 Progress Variance Monitoring.

After the meeting, if there's a delay for more than 3 days within each release, each feature lead should report to the project manager to discuss solutions and future adjustments.

3.3.2.5 Progress Variance Resolution.

After analyzing the potential reason for progress variance, for example, the quit of developers, ability inadequate of subcontract staff, incorrect estimation, etc., the project manager should re-estimate schedule and possible budget, and deal with the human resources.

3.3.2.6 FollowUp on Corrective Action.

If the problem is from the estimation, re-estimate according to the current problem.
If the problem is from the human resources, developers quit or inadequate ability, further training or end of the contract should be considered.

3.3.3 Budget Control

3.3.3.1 Cost Management.

We will use Workfront online management tool as our cost management tool. at the start of each feature, we start to sum the cost for this specific feature, and there would be a summary of cost on Friday every two weeks. Report the cost consuming condition in the meeting on Monday every two weeks to keep track of the budget.

3.3.3.2 Methods to Ensure Cost Adherence.

1. Work overtime is not recommended. However, there would be overtime pay if the work overtime is necessary.
2. There will be some bonus activities every month is the working progress is within the schedule.

3.3.3.3 Cost Control.

Start the cost record from the start of each release, so there's a subtotal cost every release. If the cost is over $4,000 per release, give a formal report to the project manager, and discuss the future cost-saving plan or budget re-estimation.

3.3.3.4 Contractor Cost Control.

We will give an evaluation of subcontract UI designers and test staff every month. If they couldn't past that, we will end the contract to ensure the developing progress and cost, and find a new subcontract as soon as possible.

3.3.3.5 Cost Variance Measurement.

$1,000 per week will be our threshold to adjust our cost.

3.3.3.6 Cost Variance Corrective Action.

We will get the reason for cost variance, and reduce the cost in the future developing process.

3.3.4 Quality Control

There will be a test for each feature and their idea. In the first two days, each feature team will gather detailed requirements they need, verification about the idea and design. There will be a formal report about this to ensure the following implementation process. At the implementation phase of the feature, there will be unit tests for each small function, and this will be done by developing a team. There will be a report about this each week to report the functioning result for each function. If there's a major defect, for example, testing not passed, report immediately to the development team, and fix within a week. After the integration of each new feature to the formal one, there will be testing done by the test team to check the integral result of the whole process. If there's a failure, notify both feature team (previous and current), and fix within 1 week.

3.3.5 Project Reporting and Communication

3.3.5.1 Electronic Media.

A computer is required for attending the formal meeting on Monday every two weeks.

A screen showing collected progress, the budget report from each feature team.

3.3.5.2 Meetings.

A stand-up meeting every day, within each feature team, take turns orally, with each person saying what has been done and what problems there are.

Formal meeting on Monday every two weeks, according to the problems mentioned in the report and collated by the document staff. Team leaders will talk about the current stage, whether to work within the schedule, budget consumption, and successes. The whole process will be

recorded by the assistants collated by them.

3.3.5.3 Information Repository.

Each member within the project will turn in a report including progress, problems encountered, their name and feature team in every release to track.

3.3.5.4 Reviews.

Documents mentioned in the above section will be collected and analyzed by assistants and analysts and the project manager. Collected problems and working status will be discussed during the formal meetings every two weeks.

3.3.5.5 Status Reporting.

Each feature team leader report the human resource moving within the project in formal meeting every two weeks.

Everyone report the current working progress in their two weeks formal report.

3.3.6 Metrics Collection

Control Chart: The manager uses this metric to control the progress of the project. Try to find out whether every release is under control.

Phase Containment Effectiveness: use these fault tracking metrics to evaluate the level of development in every release and make some improvements in future development.

CPI: The manager uses it to evaluate budgeted cost and expenditure.

3.4RISK MANAGEMENT

High risk:

1. Subcontractors can not prove equal to work.

Solution: contact subcontractors ASAP to find a solution.

Medium risk:

2. Developers quit halfway during development.

Solution: prepare extra developers for every features and requirements.

3. Computer breaks down during development.

Solution: using multiple storage ways to keep data safe.

Low risk:

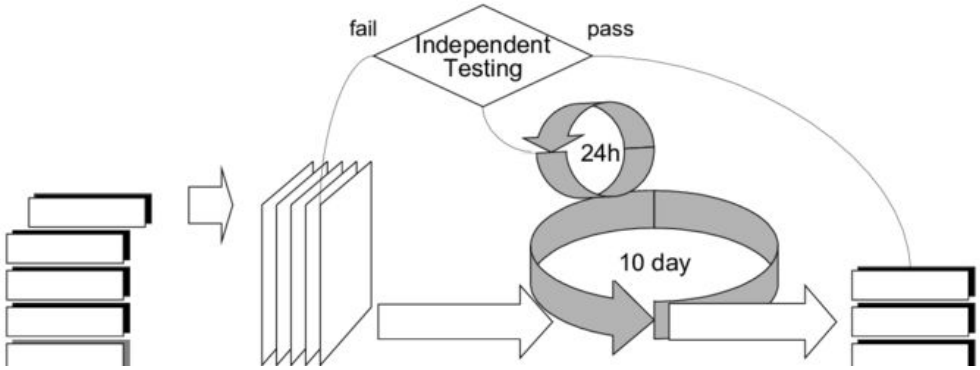
4. Staff in team catch a sick.
5. Working place catch fire

Solution: give extra time for every features and requirements.

SECTION 4. TECHNICAL PROCESS

4.1PROCESS MODEL

Our project will be using Scrum model during design and implementation, and the graphical representation of our process model is shown below:



In specific, we broke each story into one or more technical tasks. Then we iteratively do release planning, implementation and testing based on these tasks. Finally we deliver these releases as our milestones.

4.2METHODS, TOOLS AND TECHNIQUES

4.2.1 Development Methodology

In our project, we will use Agile methodology to develop our mobile application.

4.2.2 Specification and Design

We will do our specification and UI design using Axure RP 9.

4.2.3 Frameworks

In our project, we build our front-end using React Native, we build our back-end using AWS Amplify and we use Jupyter Notebook to implement and training our deep learning model.

4.2.4 Testing

For testing, we mainly focus on Automation Testing, Unit Testing, Whitebox and Blackbox Testing techniques, and we use Selenium do our test.

4.2.5 Version Control
As for version control, we will use Git.

4.2.6 Tracking
As for tracking, we use Workfront online tool to do our budget and schedule tracking.

4.2.7 Document and Management
In our project, we use Microsoft Office, Google Docs and Google Drive to do our documentation, and we use SmartSheet and Microsoft Project to do our project management.

4.2.8 Technical Standards
The technical standards we use in this project is CMM and IEEE.

4.3PROJECT INFRASTRUCTURE

4.3.1 Hardware
The hardware we use in this project is computer and smartphone.

4.3.2 Operating System
The operating system we use in this project are Windows and macOS, IOS and Android.

4.3.4 Network
We use local area networks(LAN) do our project.

4.3.5 Facilities
The facilities we use is Office building.

4.3.6 Software and platforms
In our project, the software and platforms we use including: Axure RP 9, React Native, AWS Amplify Jupyter Notebook, Selenium, Git, Workfront online tool, Microsoft Office, Google Docs and Google Drive, SmartSheet and Microsoft Project.

4.3.7 Technical Standards
The technical standards we use in this project is CMM and IEEE.

4.4PRODUCT ACCEPTANCE

1. Introduction
1.1 Purpose
This plan gives the product acceptance plan, criteria, and methodology to achieve the plan for the

Personal Food Log App and describes how the requirements will be met.

1.2 Scope
This plan provides requirements applicable to the Personal Food Log App Project. It will cover development of individual parts as well as full software assembly and checkout.

2. Responsibilities
The organization responsible for the Product Acceptance Plan will be the project managers.

3. Product Acceptance Tasks
3.1 Product Acceptance Criteria
a. Whether this app can successfully take pictures of food and store them in database.
b. Whether this app can identify food in picture and accurately calculate their calories.

3.2 Physical Configuration Audit
Based on user story.

3.3 Functional Configuration Audit
Based on user story.

3.4 Schedule
Based on schedule.

4. Resource Requirements
4.1 Hardware Requirements
We will test our mobile application using at least 100 different models of mobile phone which can cover wide range of brands in markets.

4.2 Software Requirements
We will test our software on both Android and IOS operating systems.

4.3 Documentation Requirements
None

4.4 Personnel Requirements
In our test phase, we will incorporate our team's members, customer representatives and third-party party authorities.

4.5 Test Data Requirements
We will use different size and type of images got by different cameras to do our test, and these images will also have different format and resolution.

5. Problem Resolution and Corrective Action
During the Product Acceptance activities, if our criteria cannot met, we need firstly document

these problems, then analyze the possible reason of these problems, after that we will come up with possible solutions and cost of fixing these problems and document it. We submit our documents and take action based on our report and the feedbacks.

6. Product Acceptance Environment

We need to do marketing research to rank 100 most popular smartphones, divide them into different groups based on their operating system and brands(software and hardware). Then we do acceptance test on each groups and make our report.

7. Identification of Required Artifact Evaluations

None

8. Tools and Methodologies

Tools: Smartphones, computers, simulation software, Selenium
Methodologies: Automation Testing, Unit Testing, Whitebox and Blackbox Testing techniques

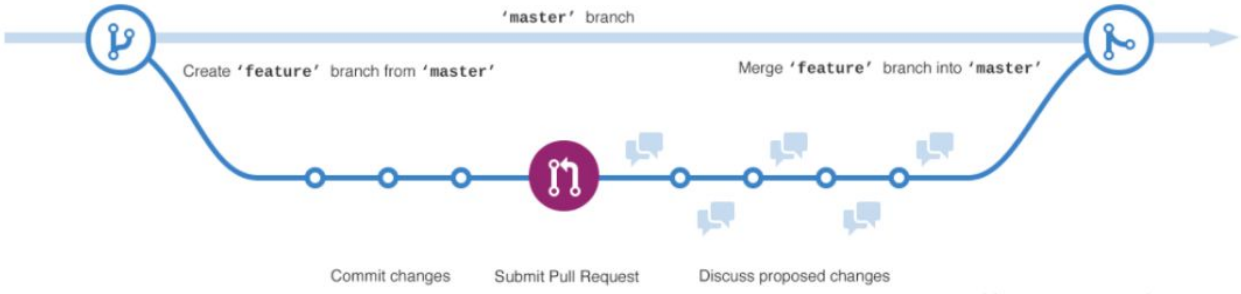
SECTION 5. SUPPORTING PROCESSES

5.1CONFIGURATION MANAGEMENT

We use Git as our configuration management tool. Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Once we obtain a workable core software, we upload it to Git and set it as master branch. Then every time we want to add new features, we create new feature branch from master to avoid influence on the stable version. After we test and make the feature bug-free, we can merge 'feature' branch into 'master' to get a new stable version.

In this way, we ensure that every version in master branch is a workable version. Once there is an error somewhere can not be solved, we can roll back to last master branch version.



5.2INDEPENDENT VERIFICATION AND VALIDATION

For verification, we do black box testing and white box testing.

When working on a feature, engineers work on this part design white box testing to make sure the new codes run well and after this part is ok, engineers use Selenium(which is a very popular automatic testing tool) to do black box testing to make sure the feature works as good as design.

Before merging feature to master branch, we have a review meeting for the feature

and our testers need to run all the features test samples so far by using Selenium automatically to make sure there is no conflict happens.

For validation, we communicate with our customers closely, every time we release a master branch version, we ask our customers to use it and tell what change we need to make or what features we need to add in our next release version.

5.3DOCUMENTATION

Document Type	Format Standard	Estimated Page Count	Peer Review Type
Requirements specifications	Non Deliverable work product	50	All agreement
Design documentation	Non Deliverable work product	40	All agreement
Traceability matrices	Non Deliverable work product	10	Manager agreement
Test plans	Non Deliverable work product	50	Manager and tester agreement
Review reports	Non Deliverable work product	20	Manager agreement
Source code	Deliverable work product (in DVD)	/	All agreement
Object code	Deliverable work product (in DVD)	/	All agreement
Host server login credentials	Deliverable work product (in book)	10	Manager and customers agreement
User's manual	Deliverable work product (in book)	20	Manager and customers agreement
Maintenance guide,	Deliverable work product (in book)	25	Manager and customers agreement

5.4QUALITY ASSURANCE

For error:

If detected when testing in feature working progress, just fix it by own and not need to report to manager.

If detected when trying to merge to the master branch, testers need to find where the fault happens, if it caused by another feature which already merge in master branch, manager needs to be told and find tech lead to have a meeting to figure out a solution like whether we need to redo the feature or rollback the master branch.

For change:

At first, we gather developers and customers together to make an agreement on whether the change is possible and suitable. Once the agreement made, we made a new feature branch from the latest master branch to process this change. After doing white box and black box testing, we have a short meeting to introduce this change to customers and managers. With their promise, engineer will merge this change feature branch to master branch.

5.5REVIEWS AND AUDITS

Requirements specifications reviews and audits: all the developers and managers and customers should join this review meeting and make an agreement for it.

Demo design reviews and audits: tech lead, UI designers and managers have a meeting for it and need to have a clear blueprint for the development.

Serious bug reviews and audits: engineers in charge and tech lead and manager need to have a meeting immediately to find out how to solve the problem.

Feature change/add reviews and audits: at beginning engineers in charge, customers and manager need to have a meeting to analyze the possibility of the change. When the feature is done, gather these people again to check whether the work is done well.

Big release version review and audits: all the developers and managers and customers should join this review meeting. Tech lead introduce every feature in this release version and everyone can show their opinions for possible change in the future or any ideas on this version.

5.6PROBLEM RESOLUTION

1. In feature development, after review meeting and dynamic testing, engineers in charge rate all the bugs found from 1-10, then solve then from high level to low level. Try to not leave any bug to next step.

2. Before merging, testers do different kinds of testing to find bugs, if the bug is found in working feature, send it back to developers to solve it. If the bug about some other features done before, find tech lead and managers to have a meeting to find out a solution.

3. If anything not clear about the requirement, should find manager and discuss with customers again.

4. If want to make any changes about the feature, find manager and get approval from him.

5.7CONTRACTOR MANAGEMENT

5.7.1 Contracting Process

Hire 6 testers as subcontractors to do unit tests for every features and requirements.

They enter team at the very first, need to join meeting at the beginning so that they start to create test plan for future testing. And every feature need testers to do automatic testing and join reviewing meeting to discuss together.

Hire 6 UI developers to design UI for the APP.

They enter team at the very first, need to join requirements and design meeting so that they can design UI plan for APP. Need to create a prototype before start coding phase. Making front-end UI for the project.

5.7.2 Contractor Performance Monitoring

For testers, manager need to check test plan before entering coding process, and rating the results of automatic testing for every feature in different phases.

For UI designers, manager need to check UI design plan and communicate with customers to make an agreement. Rating the behavior of designers' performance when facing change and new features.

If there is any problem with the subcontractors, contact the subcontractor provider to solution the problem ASAP.

5.8PROCESS IMPROVEMENT

5.8.1 Systems/Software Process Improvement Lead

1. not to make the same bugs in future development.
2. get experience about how to solve tough problems
3. handle change and features better
4. communicate with customers more smoothly

5.8.2 Systems Engineering Process Group

Every developer in this project should upgrade their coding ability and way to communicate.

The manager in this project should learn more about how to plan and manage a project and improve his ability to solve different kinds of problems.

Testers in this project should improve their ability to design test plan and test samples.