Development Plan
Software Engineering

Team 16, Durum Wheat Semolina
Alexander Moica
Yasmine Jolly
Jeffrey Wang
Jack Theriault
Catherine Chen
Justina Srebrnjak

Table 1: Revision History

Date	Developer(s)	Change
09/26/2022	Catherine Chen, Yasmine Jolly, Jeffrey Wang, Jack Theriault, Alex Moica, Justina Srebrnjak	Initial Document

The development plan details the foundation of how team Durum Wheat Semolina will develop the nutrition tracking app Utrition. This document outlines each team member's roles and responsibilities, as well as the team's workflow plan and how deadlines will be scheduled. The proof-of-concept demonstration plan addressing the biggest challenges the team will face while developing Utrition is included in this document. Also, the document lists the coding standards and technology used in developing Utrition.

1 Team Meeting Plan

Team meetings will occur twice every week. These meetings will take place from 6:30pm-7:30pm on Mondays and from 1:30pm-2:30pm on Thursdays. The location for the meetings will be on the McMaster University campus, in a private study room that will be booked and notified to all group members 24 hours before the meeting. Catherine Chen is responsible for booking the meeting room. Any additional meetings will be planned in accordance with all members' availability when needed.

All six group members are expected to attend every meeting. In the event that a group member plans to miss a meeting, they must notify all group members beforehand and continue to complete their assigned work.

Meetings will follow the meeting agenda created by Yasmine Jolly. Any member can add to the agenda which can be found in a word document inside the team's Microsoft Teams channel. Yasmine will chair all meetings, ensuring all agenda items are discussed in order of highest priority.

2 Team Communication Plan

The team will communicate primarily through Facebook Messenger chat. All members have the application downloaded on their cell phones and will receive notifications when a new message is sent. This ensures replies will be sent in a timely fashion. In emergency cases where a member must get in contact with another member as soon as possible, they may call them with their personal cell phone number. Personal contact information has been shared between all members.

3 Team Member Roles

Utrition will not employ a team leader. Team members will have equal authority and influence on major decisions and plans for the project. All important decisions must be unanimously agreed upon by all team members.

Each team member is expected to contribute to both technical development and written documentation work during this project. Tasks will be distributed throughout the span of the project where each group member is responsible for their assigned work. In addition to these responsibilities, specific roles have been

assigned to each member to ensure organization of the project and high-quality deliverables. These roles are discussed below. In the event that a team member wishes to change their role, they must discuss this matter with the team to ensure their responsibilities are being taken over.

3.1 Alexander Moica

- Backend Developer Lead: responsible for leading development of backend implementation. All needed work for product functionality must be identified and managed by this role. Questions regarding backend work will be directed to this role.

- GitHub Specialist: responsible for managing the project on GitHub including creating Git issues and managing tags. Additionally, any questions about GitHub functionality will be directed to this role.

3.2 Yasmine Jolly

- Meeting Chair: responsible for creating meeting agendas and leading all meetings. This role ensures meetings stay on track with minimal distractions and all meeting items are discussed.

- Scribe: responsible for documenting notable discussions and decisions that arise during group meetings.

3.3 Jeffrey Wang

- Team Liaison: responsible for communicating questions and concerns on behalf of the team to the class instructor and/or teaching assistants.

- LaTeXSpecialist: responsible for answering all team internal questions related to LaTeX.

3.4 Jack Theriault

- Testing Lead: responsible for leading testing processes and ensuring other team members are creating complete test cases for their code. Final testing efforts will be done by this role.

3.5 Catherine Chen

- Frontend Developer Lead: responsible for leading development of frontend implementation. All needed work for user interface implementation and design must be identified and managed by this role. Questions regarding frontend work will be directed to this role.

- Meeting Room Booker: responsible for booking private study rooms on McMaster University's campus for team meetings. This role will book a room and notify all team members 24 hours before the scheduled meeting.

3.6 Justina Srebrnjak

- Documentation Proof Reader Lead: responsible for leading reviews of all documentation. This includes ensuring the document is cohesive, correcting any grammar or spelling errors, and grading the document with provided rubrics before submission. Any questions related to proof reading documents will be directed to this role.

4 Workflow Plan

The team will employ a **Trunk-based Development** version control model, which encourages small and frequent merges to the main branch. This model will help facilitate the use of CI/CD technologies throughout the product's development. Developers will create individual branches when making their changes, and create a pull request when changes are ready for review. Once a pull request is opened, Git Actions will run automated tests and use linting tools to flag potential programming and stylistic errors. Once the changes pass code checks and receive an approval from another team member, the changes will be ready to merge into the main branch. Under this model, changes to the product will be added and reviewed, quickly and consistently.

To manages issues, weekly meetings have been designated as 'planning' and 'update' meetings. Management of issues will primarily be covered in planning meetings. Members will create issues on GitHub, representing a specific aspect of the project they are completing. Members will by assigned to issues pertaining to the changes they are making. These issues will be tracked on the Utrition Project Board, where issues will be placed in categories such as To Do, In Progress, Review, QA, and Done. Issues will be linked to specific pull requests to better communicate the purpose of each pull request. The creator of the pull request will also add any appropriate tags (ex. documentation, bug fix, etc.) to better categorize the work and locate easily.

The full workflow is as follows:

1. Create issue on GitHub project board
    (a) Assign issue to self
    (b) Fill out issue information
2. Pull changes from `main` branch
3. Create new branch off of `main` branch
4. Commit changes to new branch
5. Open pull request
    (a) Add description, appropriate tags and link to issue

    (b) Changes must pass automated tests

    (c) Changes must pass linting checks

    (d) At least one other team member must approve changes

6. Merge changes into `main` branch

5 Proof of Concept Demonstration Plan

Utrition is a complex application that contains risks to the completion of the project. These risks are learning how to utilize the TensorFlow library, training the machine learning models to accurately recognize and classify the images presented, and learning frontend languages such as HTML/CSS and the backend connector language, JavaScript.

The proof-of-concept demonstration for Utrition will consist of three main components to accurately depict the feasibility of the project. There should be a simple user interface that the backend implementation will be accessed through. In terms of backend functionality, there should be simple artificial intelligence. This technology will be receiving an input of food items and should be able to correctly output the identity of the food displayed in the image for a small subset of our selected food list. Upon the correct identification of the food, the application will output the corresponding nutrition facts.

In the case that the risks are not able to be mitigated in the proof-of-concept demonstration, the project will readjusted. For example, if it proves too difficult to create the frontend of the web application, the product will be accessed from the backend files using command line.

6 Technology

The following technologies will be used for the development and testing of the system:

- Programming languages

  - Python
  - JavaScript
  - HTML
  - CSS

- Linter tool

  - Pylint

- Unit testing framework & code coverage measuring tool

  - Pytest

- Libraries
  - NumPy
  - TensorFlow
  - Pillow
  - OpenCV-Python
  - Pickle
- Development tools
  - Visual Studio Code
  - Git
  - Chrome web browser
  - LaTeX
  - GitHub Actions (CI/CD)
- Remote workspace & video conferencing
  - Microsoft Teams

For continuous integration, automated unit tests and linting will be run each time a branch is being merged. This will ensure that each revision that is committed will trigger the automated tests. The working copy should pass the automated tests before merging into the main branch.

7 Coding Standard

Team Durum Wheat Semolina will adhere to the PEP8 Python coding conventions by using Pylint. When a team member has made changes to a Python file, Pylint will be used before submitting a pull request. For code in JavaScript, HTML, and CSS, the team will be following the Google style guide. The hyperlinks to the guides can be found below:

- Python
- JavaScript
- HTML / CSS

8 Project Scheduling

Project scheduling will occur during the two weekly meetings mentioned in Section 1 - Team Meeting Plan. Monday meetings will be "planning" meetings where the team discusses and breaks up upcoming tasks into specific actionable issues, create tickets on our GitHub board, allocate work to team members,

and determine goals and deadlines for our project deliverables. During Thursday meetings, the team will be checking the progress of every ticket assigned to a team member and discussing any issues that have arisen in the interim. Through this system, we are able to effectively schedule deadlines and ensure we meet them through constant communication, progress checks, and efficient distribution of work.