SFWRENG 3XA3 L03
Development Plan

SaveTheDate
Team 17

Karuka Khurana (khurak1)
Utsharga Rozario (rozariou)
Samarth Kumar (kumars38)
Dhruv Cheemakurti (cheemakd)

February 4, 2022

Contents

List of Tables

List of Figures

Table 1: Revision History

Date	Developer(s)	Change
Feb 4, 2022	Karuka Khurana, Utsharga Rozario, Samarth Kumar, Dhruv Cheemakurti	Creation of Development Plan Document

1 Team Meeting Plan

a. Meeting Location and Time
   Group 17 will be meeting in person at the Information Technology Building on campus in room 236. This will start the week of February 7th, 2022, every Monday and Wednesday during the SFWRENG 3XA3 lab session (L03). Should the team agree that additional meetings need to be held, the meeting lead will arrange a time and location that suits all team members.

b. Meeting Agenda and Roles
   Each meeting will be structured with each member having a meeting role. These roles are outlined in Table 2. The meeting manager oversees the meeting and ensures it runs smoothly by guiding the rest of the team members through the agenda. Table 3 describes the rough outline of the agenda that each meeting will have. The note taker will be in charge of taking meeting minutes and noting all important action items and main points discussed during the meeting. The progress manager must update the Gantt chart at the end of each meeting to keep track of the project progress. This will help the team understand what has already been done and what still needs to be done. Finally, the summarizer will be responsible for summarizing what was discussed in the meeting to the rest of the team so that everyone is on the same page and assign tasks that need to be done before the next meeting. These roles will switch every week, which means each member is responsible for each role twice a week. Karuka will start by being meeting manager, Samarth will be note taker, Utsharga will be the progress manager, and Dhruv will be the summarizer. Each member must be responsible for all 4 roles before repeating one. Should a member be unable to attend a meeting, their role will be assigned to one of the present members.

Meeting Manager	Confirms the attendance of all members and ensures the meeting is running as per the meeting agenda
Note Taker	Responsible for noting important changes and action items discussed during meetings
Progress Manager	Responsible for updating the Gantt chart and recording progress on deliverables
Summarizer	Summarizes the main points of meetings and informs the group on what needs to be done by the next meeting

Table 2: Team Meeting Roles

Step 1	Attendance taken by meeting manager
Step 2	Meeting manager outlines meeting objective
Step 3	Discuss status on tasks assigned from previous meeting
Step 4	Issues and obstacles that need to be discussed and worked on
Step 5	Summarizer assigns tasks that need to be completed for next meeting
Step 6	Summarizer discusses objective for the next meeting
Step 7	Ensure meeting notes are complete

Table 3: Meeting Agenda Rough Outline

2 Team Communication Plan

Besides the conversations taking place during lab time, the team will communicate through Facebook Messenger. This chat is there to let group members know if one cannot attend meetings, has any questions or any concerns. A Discord channel will be used to host virtual meetings and our Gantt chart will be a visual aid in understanding our progress throughout the term. The chart will include deliverables, their deadlines, and their status on completion.

3 Team Member Roles

Table 3 outlines the roles each member will hold for the duration of the project. These were chosen based on the comfort level of each member in their respective role.

Project Manager	Utsharga Rozario
Software Developers	Karuka Khurana, Dhruv Cheemakurti, Samarth Kumar, Utsharga Rozario
Documentation Lead	Karuka Khurana, Dhruv Cheemakurti, Samarth Kumar, Utsharga Rozario
Gitlab Lead	Dhruv Cheemakurti
LaTeX Lead	Karuka Khurana
Technology Lead	Samarth Kumar

Table 4: Team Member Roles

4 Git Workflow Plan

A remote Git repository will be made so that each team member can clone it and have a local version on their machine. The members are expected to regularly pull from the repo to avoid confusion as well as push the changes they have made. In the case of members working on the same code, each member would work on their separate branch to avoid conflicts. The members would then merge each branch into the master so that it always holds a working error free version of the application. Labels are used to show the issues, bugs and improvements in the development process and are given to each team member to fix them.

5 Proof of Concept Demonstration Plan

5.1 Difficulties and Concerns

1. **Implementation:**
   The main implementation is a PDF scraper using Python. Group members may not have experience working with scraping tools, so this may pose a challenge. Furthermore, generation of a dashboard using React-Notion is unfamiliar to the group and may be difficult.

2. **Testing:**
   Testing will be performed to ensure scraping functionality works and that the dashboard is generated effectively. This may be difficult because it will require sufficient sample cases to show that date/deadline extraction works properly. It will also require qualitative testing to ensure the front-end looks appropriate.

3. **Required Libraries:**
   Our group will use three Python libraries for PDF scraping and manipulation. The re library allows for the use of regular expressions, which can be used to find important identifiers in the text. The tabula-py library allows for structured and unstructured reading of the PDF files. The pandas library is used for data manipulation and reshaping after extraction (using DataFrame, pivot, etc.). The main difficulty will be learning how to use these libraries for the first time. For the scraper to work on any course outline, unstructured reading using tabula-py may be necessary in some instances, which is more difficult to implement than structured reading.

4. **Portability:**
   There should be no difficulties with portability because a web application is being developed using Notion written in React, which is compatible with the widely used web browsers (Chrome, Edge, Safari, etc.).

5.2   Overcoming Risks

1. **Implementation:**
   Group members have prior experience using Python in past courses or projects, so the language is familiar. Since Python is such a widely used language, documentation is readily available throughout development. The group will look to existing PDF scrapers for a better understanding of how to adapt them to fit the desired purpose. Risks to dashboard generation can be overcome through looking at the existing React-Notion code and through trial and error.

2. **Testing:**
   By using frameworks such as pytest and Cypress, risks of testing are overcome through automation of some or all parts of the implementation. Additionally, the team can pool together past course outlines as a starting point for testing scraping. More rigorous testing including edge cases can be performed by generating fake outlines with various and tricky date/deadline formats.

3. **Required Libraries:**
   Overcoming the challenges with the desired libraries can be done through reading the documentation and going through web tutorials. There is also regular expression "cheat sheets" that can be used to ensure various date formats can be identified in the PDF. Since PDF scraping is widely used, additional libraries may be found that can accomplish our goals in an easier or more efficient fashion.

5.3   Initial Demonstration Plan

As an initial demonstration, PDF scraping functionality should be implemented for at least one simple course outline, to demonstrate that dates can be extracted from the document. This also demonstrates that the risks of implementation, including familiarity with the programming language, are overcome.

6   Technology

Existing code is written in React and TypeScript. The PDF scraper will be implemented using Python and libraries including pandas, re, and tabula-py. VS Code will be used as the IDE. Code will be documented using Doxygen. The pytest framework will be used to assert that the scraper catches all important dates or deadlines. Cypress will be used to test the front-end, with more testing details to be discussed in the future.

7   Coding Style

Due to the extensive use of React.js for this project, it is best we follow The Airbnb JavaScript Style Guide for React.js, due to its popularity. In addition, Airbnb styling emphasizes documentation and makes the code more understandable for developers, and with the usage of the compatible linting tools we can ensure this style guide is followed throughout the project. Similarly, for the use of Python in this project it is best to follow the PEP 8 format of code styling. PEP 8 helps make the code more readable and has many compatible linting tools to ensure that the PEP 8 standard is held.

8   Project Schedule

This following link contains detailed information about our project scheduling and our Gantt Charts:
https://gitlab.cas.mcmaster.ca/se3xa3_l03_g17/se3xa3_l03_g17/-/blob/main/ProjectSchedule/3XA3_L03_G17_GanttChart.pdf

9   Project Review

N/A