FIT3163 - Data Science Project 1 - Assignment 4 - Project Design

Workshop: Friday, 2pm - 4pm

Group: FIT3163_CL_04

Members: Elaine Liong (ID: 29942357),

Jack Ooi (ID: 29037077),

Vionnie Tan (ID: 30092809)

Table Of Contents

Introduction

The purpose of this document is to outline how our final product will look like. This will include 2 high level designs relating to the online tool that we are developing. UI interface mockup diagram will show how our final product will look like and a flowchart will explain the relationship between the components in our final product. This document will also include the software environment and hardware required by the system, as well as a brief justification of our choices and decisions made.

1. Design Representation

1. UI Interface Mockup Diagram

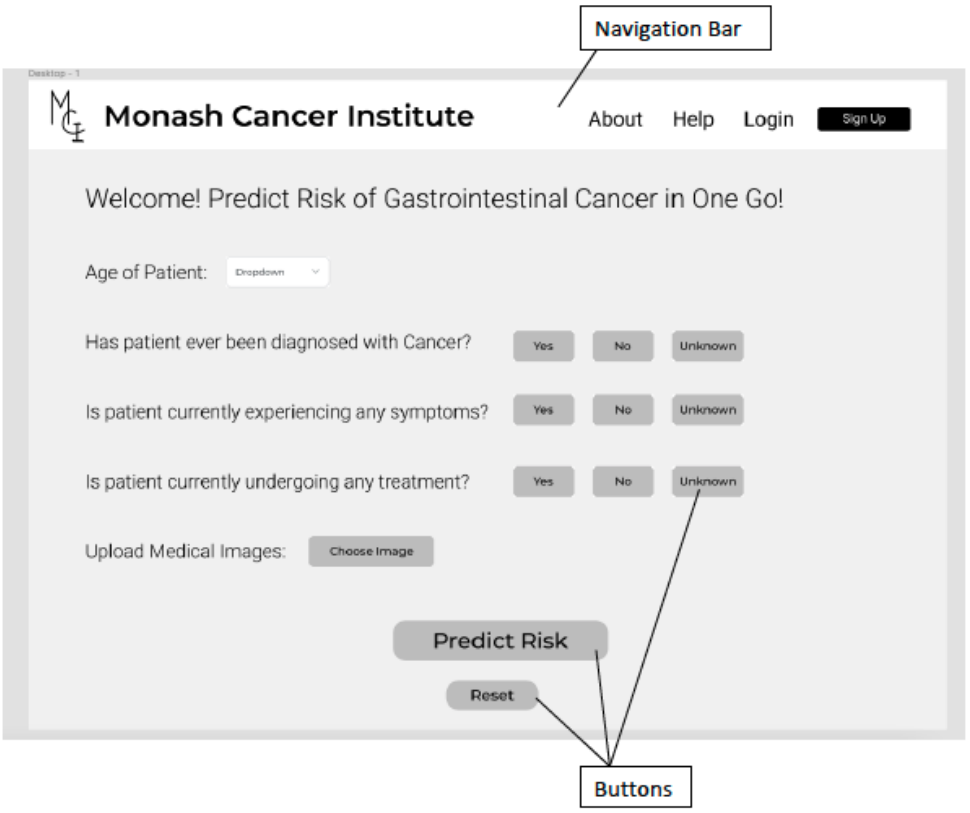

Figure 1: Landing Page Mockup of an Online Tool for Cancer Prediction

The UI Interface Mockup Diagram was chosen as our design as we plan to deploy our predictive model on an online tool for health officials to use. The design was made using the website “Figma”, and represents what our website will look like. On our main page above, there are buttons alongside with questions that patients can fill. Buttons and the navigation bar on top are shown clearly. This is the front-end design of our project, and our next design,

the flowchart shows the back-end processes of our website and describes how each process integrates with one another.

2. Flowchart

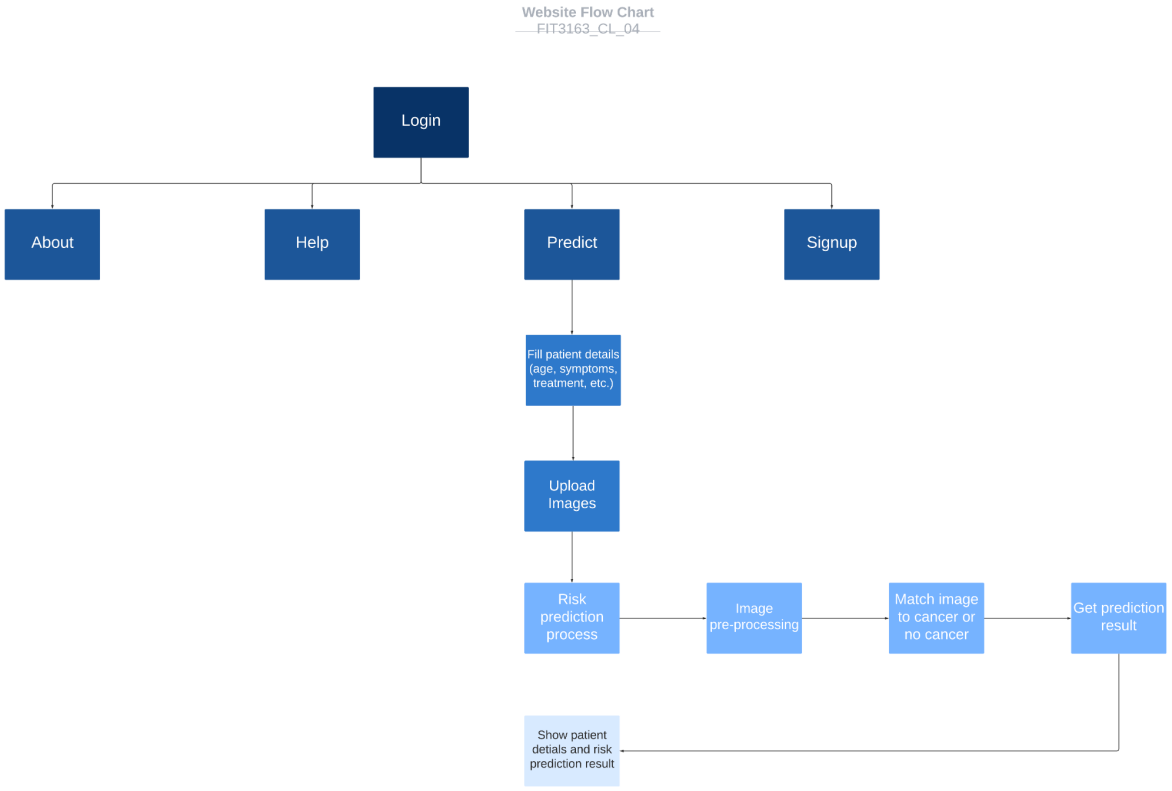

Figure 2: Online Tool for Cancer Prediction Flowchart

A flowchart, created with Lucidchart, is chosen as our second design representation to display the sequence of activities visually. In the main page of our online tool for cancer prediction, users will be required to login to be able to use the cancer prediction functionality. If the user is not registered, they will be able to sign up. Users will also be able to see what our online tool is about as well as ask for help if they need to. In using the predict functionality, the users will be required to fill in their details and questions shown in Figure 1, upload their medical image, and click the predict risk button which will then start the risk prediction process. The risk prediction process includes image pre-processing, classifying or matching the pre-processed image to cancer or no cancer, and obtaining the

prediction result. The result along with the user's pre-filled details will then be displayed to the users.

2. Software and Hardware Specification

Specialised software

- Website design:
    - Figma
    - Lucidchart
- Website: WordPress

Software libraries

- Python:
    - Matplotlib
    - Seaborn
    - Numpy
    - Keras
    - Sckit-learn

Operating softwares

macOS and windowOS

Programming language environment

- IDE: PyCharm
- Programming language: Python

Hardware requirements

- Laptop or Desktop Computer

Computing requirements

- 16 GB of RAM
- Medium-performance based CPU & GPU

Data storage requirements

- University Google Drive Storage
- AWS Services

Networking requirements

- Network Connectivity of minimum 5MB/s

Project management tools

- ProjectLibre
- Revision control system: GitHub
- Code bug tracker

3. A Justification of Choices

Specialised software

- Website Design:

  Figma is chosen as a tool to design our user interface since it is a relatively simple web-based design application. Using this application, we are able to create a mockup of our online tool homepage with ease which gives an overall idea of how our final product will look.

  Our team also chose Lucidchart to visualise the backend flowchart of our online tool because it allows team members to easily collaborate on drawing and revising the flowchart. As an alternative to Lucidchart, we have considered using Microsoft Visio. However, Lucidchart is easier for the team members to access as it is web-based and the team members are also more familiar with Lucidchart, so we decided to use Lucidchart instead of Visio.

- Website

  Our team chose WordPress to build our website since it has a shallow learning curve which is beneficial for us as we have no prior knowledge in website-building. It is also free and open-source. Additionally, most functions that we require can be easily implemented by plug-ins. We have also considered using CakePHP framework to build our website. However, this option might be unworkable for us since it has a steep learning curve as our team members are not familiar with using the PHP language.

Programming language and software libraries

Python is chosen for the programming language since all members are familiar with it and have background knowledge of python. The libraries used to create the model are:

- Matplotlib - able to create interactive visualizations for users (users are able to check the visualizations for their cancer result).

- Seaborn - extension for matplotlib for drawing more informative statistical graphics (more informative graphics for users).
- Numpy - backend programming library used for process data, mathematical operations on array and matrix data structures.
- Keras - deep learning API to develop deep learning models, allow to define and train the model efficiently using the libraries wrapped.
- Sckit-learn - used for machine learning, have efficient tools for classification, regression, clustering, etc to help train the model.

As an alternative to Python, we have considered using R as our programming language as all of us have previous experience in predictive modelling using R. However, uncertainties regarding performance constraints as well as the lack of packages that are available in comparison to Python have led us to decide to use Python as our software library. For example, python has many more packages for data modelling and data visualization (scikit-learn, numpy, matplotlib and seaborn), however, for R, the only available options are ggplot2 and tidyverse.

Programming language environment

Our team has chosen PyCharm as our IDE for Python as it offers the best package across other IDEs and prior experience from using it has shown it to be of great reliance. Alternatives such as Anaconda was considered, but since Anaconda uses an online interpreter, we decided to use PyCharm.

Hardware requirements

Laptop or desktop computers are required in our team project. It is important that our team has their own machine to work with so that our project can be finished on time. Laptops are chosen when the situation requires mobility, desktops are chosen when the situation requires performance.

Data storage requirements

An extensive research into the possible storage requirements for our projects have led us to determine that we would need at least >10GB of storage for our project. As university students, we are given with free Google Drive storage for us to use. We plan to use this to store generated models as well as for the given dataset of images. If Google Drive proves to be unreliable or unusable due to unforeseen circumstances, we have selected AWS as an alternative. AWS is known for its reliable services, but a downfall would be that it requires us to pay them a fee to use their services.

Networking requirements

Throughout the course of our project, we'd have to collaborate with peers and conduct regular meetings. To do this, it is important to have reliable and fast internet connections to foster the smooth communication between team members especially since everything has to be done online.

Computing requirements

As we are dealing with image processing, it is very important for us to have fast processing times and to do that, we would need:

- 16 GB of RAM to run the website and image processing algorithm faster and more effectively.
- Medium-performance based CPU & GPU that allows training and testing across more images faster which leads to faster models being developed. Also, able to display the graphic with ease.

Project management tools

- Revision Control System: GitHub

  As most of us would be working on our own devices at our own times (due to distance constraints), we decided to use GitHub to allow for regular project management as GitHub allows ease of upload and download of files which can be accessed by all of our team members. Using GitHub, we will be able to code together as a team, and share feedback for each other's code. Furthermore, storing our code

online on GitHub's servers minimizes the chance of losing the actual code. One constraint would be that GitHub can take a while for files to be uploaded and viewed. GitHub's public nature could raise privacy concerns as other teams would be able to view our uploaded code.

- Projectlibre

  Projectlibre is needed in our team project for project management. It allows us to record down every important decision or steps of the project and allows us to track it back so we could revise what we did in the previous steps. Besides, it also allows us to record the start, finish date and duration of a certain decision or steps, in this way, it allows us to finish up the project more efficiently and clearly.

- Code Bug Tracker

  A code bug tracker is needed for our team to record and track the bugs in our predictive model as well as our online tool where our predictive model is deployed. There are a few options of code bug tracker, such as Backlog, Clubhouse, and GoodDay, that are in our consideration. Each of them has their own pros and cons. After some considerations, Clubhouse seems like a suitable option for our team as it supports features that support Agile project management concepts like scrum sprints and burn down charts.