



Middle East Technical University Northern Cyprus Campus
Computer Engineering Program

CNG491 Computer Engineering Design I

Iris Analyzer

2269421 Aref Khademi Najafabadi
2246452 Sameh Farid S. Algharabli
2269454 Evans Muthugumi Kimathi
2175263 Elbaraa Elsaadany

Supervised by
Asst. Prof. Dr. Meryem Erbilek

Report No.1

Contents

1	Introduction	3
1.1	Motivation	3
1.2	Aims and Objectives	3
1.3	Methodology	3
2	Requirements	5
2.1	Stakeholders	5
2.2	Functional System Requirements	5
2.3	Non-functional System Requirements	6
2.4	Domain Requirements	6
2.5	Assumptions and Justifications	6
3	System Modelling	7
3.1	Structured Use Case Diagram	7
3.2	Sequence Diagram of the Major User Cases	7
3.3	Context Model	7
3.4	Architectural Model	8
3.5	Process Model	8
4	(Graphical) User Interface	13
5	Agile Development with Scrum	14
5.1	Sprint Backlog	14
5.2	Sprint Burndown Chart	14
5.3	Sprint Review	15
5.4	Sprint Retrospective	16
6	Estimation	17
6.1	Input	17
6.2	Output	17
6.3	Query	17
6.4	Logical Internal Files	17
6.5	External Interface	18
6.6	Total Unadjusted Function Cells	18
6.7	Influence Multiplier	18
6.8	Total Adjusted Function Calls	18
6.9	Line of Code	19
6.10	Estimate the Efforts	19

Chapter 1

Introduction

1.1 Motivation

Iris recognition can be used for verifying owner identity (biometric data) and demographic data of the identified person (soft biometric data). The main motivation for this project is to provide a better way for identification of humans and their demographics. It also offers more efficiency because it is harder to steal or replicate human eyes than it is for fingerprints and other biometric techniques.

any available study in this field? it will be different?

1.2 Aims and Objectives

The main aim of this project is to acquire identification and their demographic information by analyzing their eyes (iris). The objectives are:

1. Get eye data from a database.
2. Get the geographical and texture features from the eye.
3. Use the features acquired to retrieve the identity and demographic data.

1.3 Methodology

The project will be done in two parts: image processing part and machine learning part. The image processing part will be achieved using the following methods:

1. Acquiring the eye image from the database.
2. Segmentation of the eye image to retrieve the required parts of the eye (iris and pupil).
3. Feature extraction which will entail extraction of both geometric and texture features from the segmented part.
4. Feature correlation which is the removal of the unwanted and redundant features acquired from feature extraction.

The machine learning/deep learning part will use the methods below:

1. Classification of data acquired into relevant categories.
2. Prediction and learning – the system will have to learn and output the correct data (identification and geometric data) from the classification step above.

Chapter 2

Requirements

2.1 Stakeholders

1. System administrator
2. Application users

2.2 Functional System Requirements

1. The user shall be able to choose eye images as inputs.
 1. The system shall display eye images as input to the screen.
 2. The system shall mark all the eyes images selected by the user as wanted input.
 3. The system shall move copies of the chosen eye inputs to the next stage (processing stage).
2. The user shall be able to choose the expected output as identification data.
 1. The system shall display identification as one of the outputs.
 2. The system shall check if the identification output has been chosen.
 3. The system shall use the chosen input to process the data.
3. The user shall be able to choose the expected output as demographic data.
 1. The system shall display demographic as one of the outputs.
 2. The system shall check if the demographic data output has been chosen.
 3. The system shall use the chosen input to process the data.
4. The system administrator shall be able to edit the eye images in the database.
 1. The system shall display an edit confirmation message to the system administrator.
 2. The system shall display a list of edit options to the system administrator.

3. The system shall display a save edits made confirmation message to the administrator.
4. The system shall save the edits if the confirmation message is confirmed.
5. The system shall discard the messages if the confirmation message is declined.

Modelling?

2.3 Non-functional System Requirements

1. The system shall be available to users 24hrs a day.
 1. The system shall check every 2hrs (system troubleshoot) that all parts are functioning properly.
 2. The system shall send a notification to the system administrator about any errors that occurred.
2. The system shall work on windows operating system. → only?
 1. The system shall run on all devices with windows 7 and above operating system.
 2. The system shall display an error message if the device is incompatible.

3. The system shall be coded using python programming language.
 1. The system shall be built using python programming language.

2.4 Domain Requirements

1. The user shall choose only one eye image as an input.
2. The user can choose either one output option (identification or demographic data) or both outputs to be displayed as the result.

2.5 Assumptions and Justifications

1. The eye images used are not captured in real time but are taken from a database. These is due to the following reasons:
 1. It needs special cameras and scanners.
 2. The eye is sensitive to flashes and they might be damaged.
 3. Ethic and privacy concerns, because users may not be sure of their safety of their images.

Move it to the func. req.

Chapter 3

System Modelling

3.1 Structured Use Case Diagram

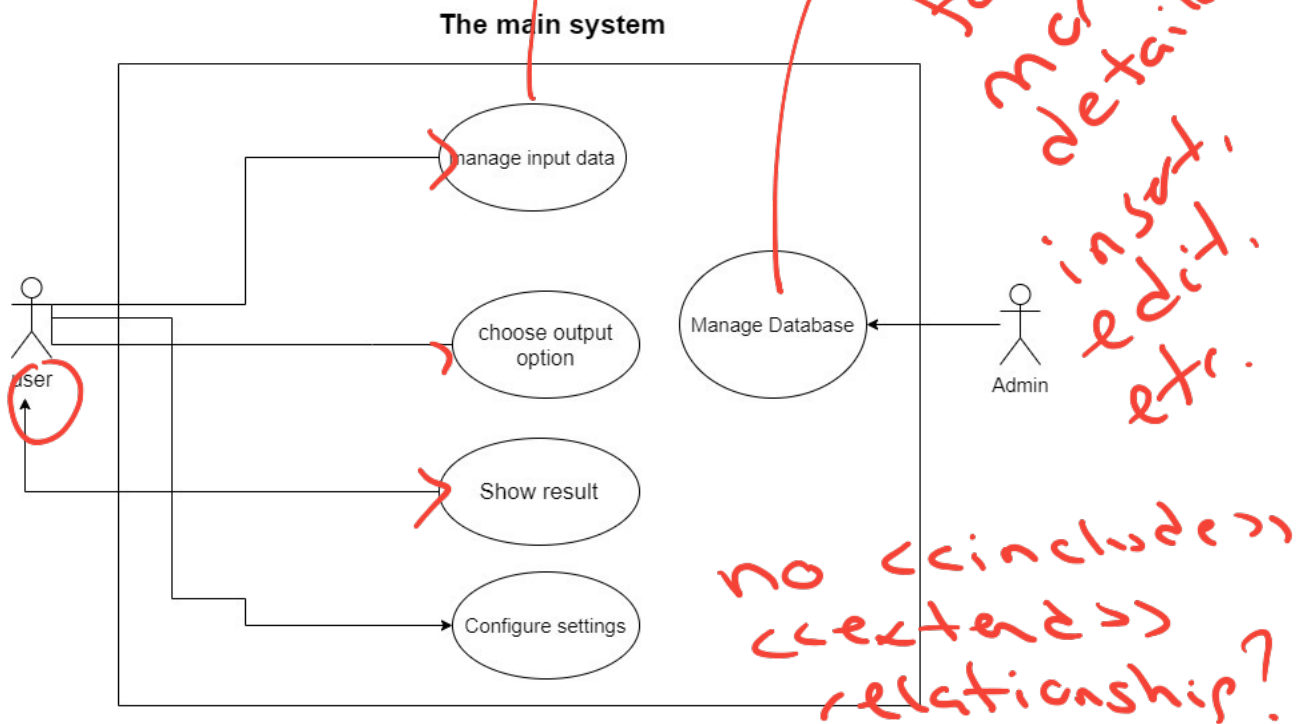


Figure 3.1: Structured Use Case Diagram for Iris Analyzer System

Modelling?

3.2 Sequence Diagram of the Major User Cases

3.3 Context Model

For the context model we have no external systems.

but you have users!
what about DB?

Input → Processing → Output
7 optional button alt

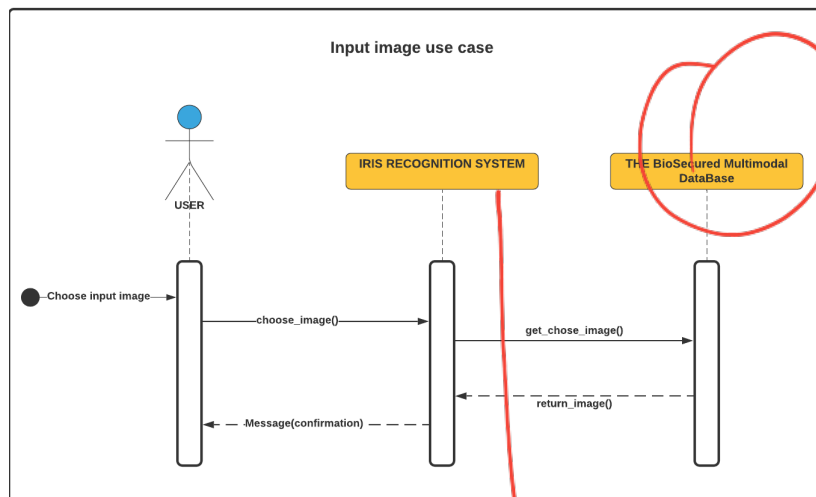
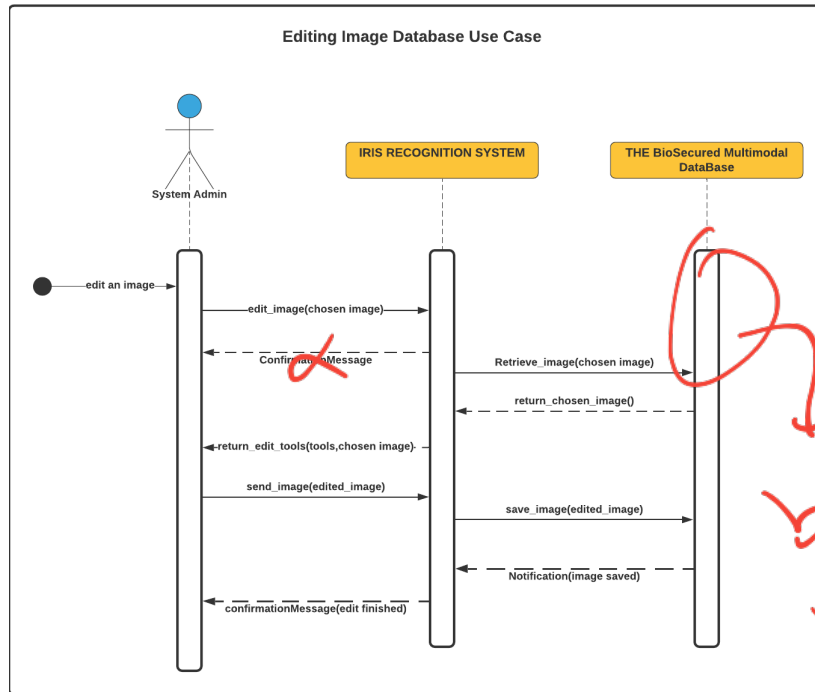


Figure 3.2: Input Image Use Case

3.4 Architectural Model

3.5 Process Model



be careful with activation boxes

Figure 3.3: Editing Image Database Use Case

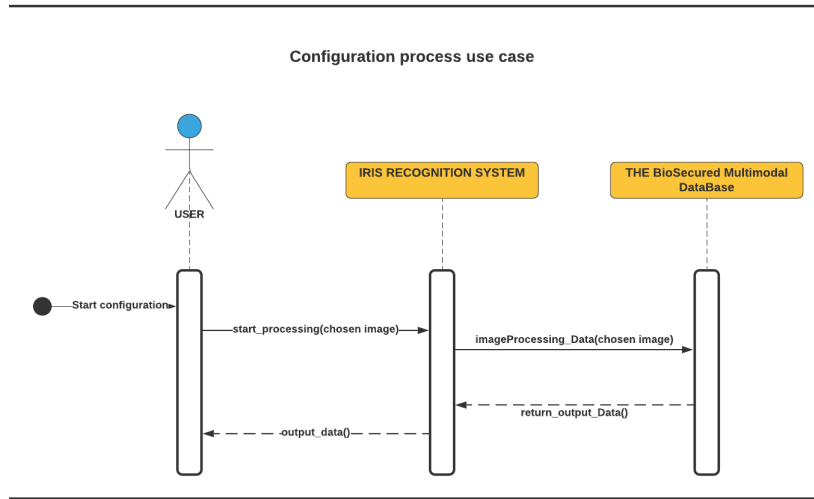


Figure 3.4: Configuration Process Use Case

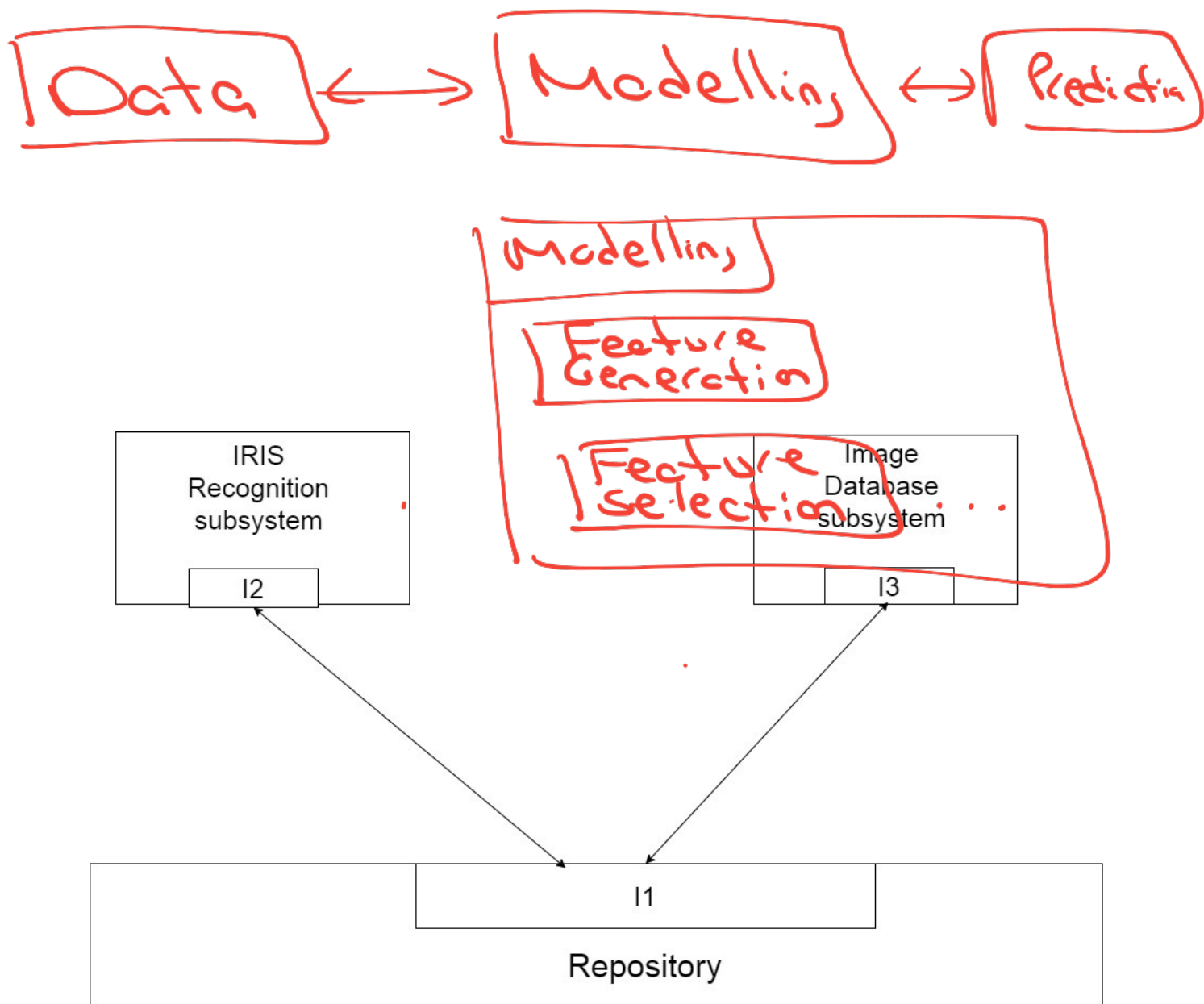
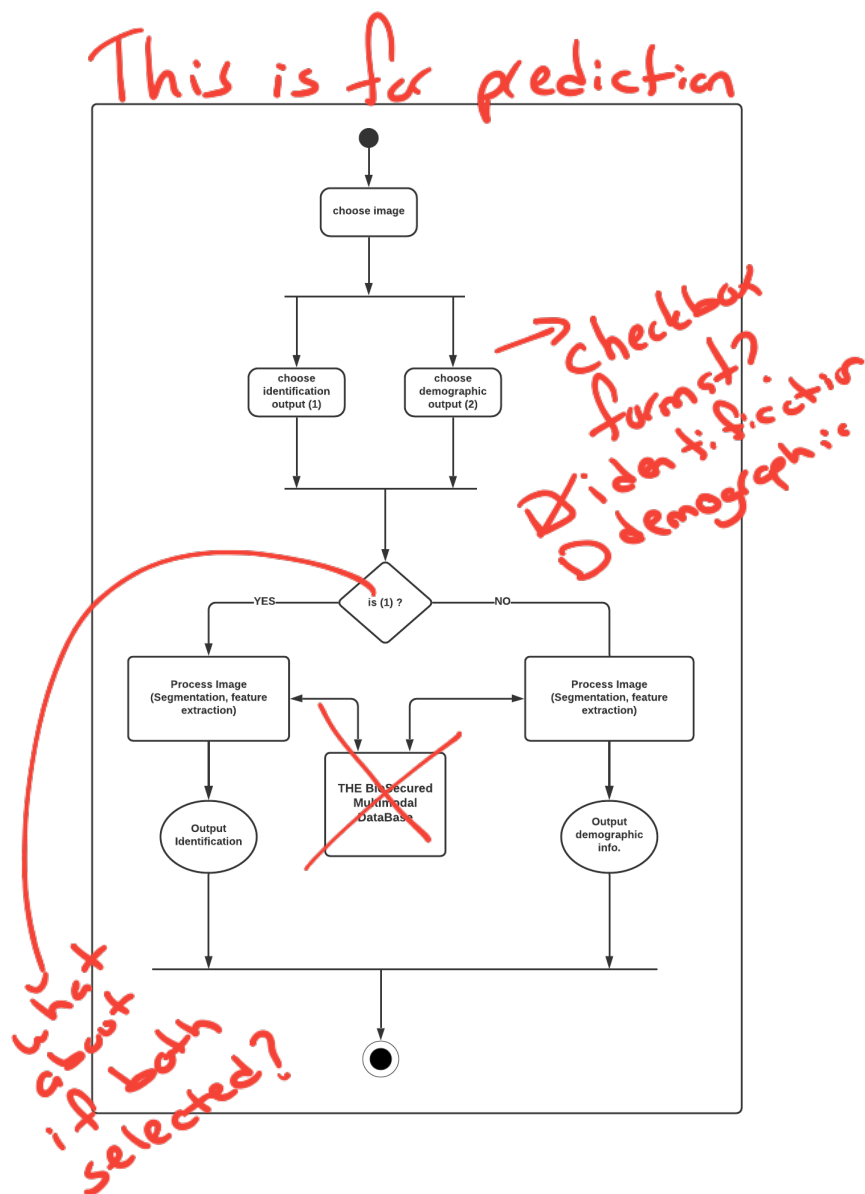


Figure 3.5: Architectural Model for the Iris Analyzer System



You should have another one for modelling.

Figure 3.6: Process Model for the Iris Analyzer System

Chapter 4

(Graphical) User Interface

Below is a very early iteration of how the graphical user interface of the Iris Analyser application will look like. The interface is in early stages of design and is subject to many changes across the upcoming sprints.

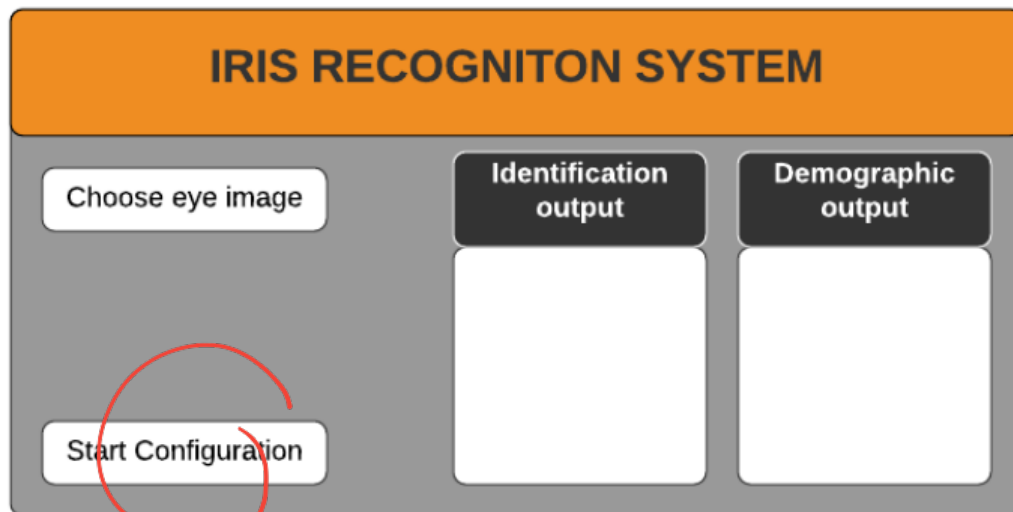


Figure 4.1: Graphical User Interface v.0.0

→ GUI?
You should add explanation
for each GUI as well.

Chapter 5

Agile Development with Scrum

5.1 Sprint Backlog

who is responsible for what?

Table 5.1 shows the list of tasks completed so far and the time spent on each one. It also includes the priority leave for each task.

Table 5.1: Sprint Backlog table

	Sprint Backlog Tasks	Priority (High, Med or Low)	Time Spent (Hours)
1	Research (Image Processing techniques)	L	6
2	Write a Literature Review	L	3
3	Research (Deep Learning basics)	L	7
Sprint 1 Gather enough information regarding deep learning in Iris Recognition studies			

5.2 Sprint Burndown Chart

Table 5.2 shows the remaining time of the tasks of the first sprint over the past 3 weeks against a general sprint guideline.

Table 5.2: Sprint Burndown table

Task #	Sprint 1 Backlog Tasks	Week 1	Week 2	Week 3
	Guideline	16	8	0
	Remaining Values	16	10	2
1	Research (Image Processing techniques)	6	0	0
2	Write a Literature Review	3	3	0
3	Research (Deep Learning basics)	7	7	2

Figure 5.1 shows the graphical format of the above Sprint Burndown chart with the red line indicating the guideline to follow, and the green line representing the group progress on the hours spent and remaining on the tasks.

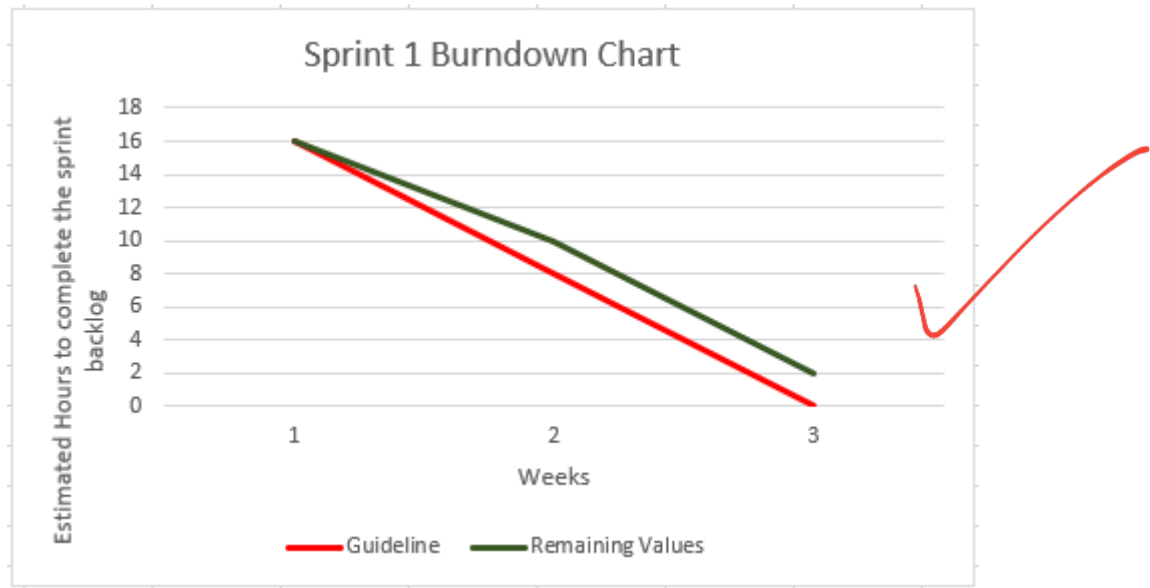


Figure 5.1: Sprint Backlog graph

5.3 Sprint Review

In order to determine if the current sprint (sprint 1) has been successful and has met its goal we need to have a look at the table of information provided above. Although, at first glance it may be assumed that the sprint was unsuccessful since the estimated hours of work remaining is not zero, however, this is not the case. This sprint has been a short one and is the first sprint in the agile development process, therefore, there have not been detailed nor quantitative tasks to complete in this short time period and hence, being behind by a few hours is no reason for worry.

Problems and issues during this sprint have been prevalent during this sprint. The graphical data may show that progress line is close to the guideline, however, there have been problems in this period. The first problem we encountered as a team was a result of poor communication between the group members and the supervisor. The weekly meetings left many topics and questions regarding the project itself unclear as the team as a whole did not have a clear vision on the goal of the project and its steps. However, after setting up an extra semi-daily group meeting, constant clarifications were being made and project started to take shape nicely.

5.4 Sprint Retrospective

In retrospect, the sprint so far has been a success on paper, but has place for improvement. With time, the sprint will take shape as more tasks are added and with the project steps becoming clarified and the direction in which the team will move on with the project is finalized.

In order to further improve the sprint, we are planning to incorporate our supervisor more strongly in the agile development process. In addition to this, more improvements can be made by updating the sprint backlog and sprint burndown chart more consistently and by showing it in the weekly meetings as well as semi-daily meetings.

Chapter 6

Estimation

6.1 Input

Each input in GUI is 1 functional point in which some are complex, simple and average.

Input	Complexity	Multiplier	Functional Points
Register user	Low	3	1
Log in	Low	3	1
Manage Data	High	4	1
Request Image Result	High	1	1
Manage settings	Medium	6	1

6.2 Output

In the system, there are output GUI that is counted as 1 functional point, in which some have a medium complexity depending on the GUI.

Output	Complexity	Multiplier	Functional Points
Images	High	1	1
Image data	High	4	1

6.3 Query

Query	Complexity	Multiplier	Functional Point
Get recent activities	Medium	4	1

6.4 Logical Internal Files

In the system there are 3 main tables that have transactions that each is counted as one functional point.

Logical internal file	Complexity	Multiplier	Functional Point
Image Data Table	Medium	10	1
Feature Table	Low	7	1
Image table	Low	7	1

6.5 External Interface

Samba is the Linux implementation of the SMB/CIFS file sharing standard used by Windows PCs and Apple computers

External interface	Complexity	Multiplier	Functional Point
	High	10	1

→ why do you need it?

6.6 Total Unadjusted Function Cells

Total unadjusted function cells	91
---------------------------------	----

6.7 Influence Multiplier

TDI	1- data communications	4
	2- Distributed Functions	3
	3- Performance	5
	4- Heavily used configurations	1
	5- Transaction Rate	2
	6- online data entry	3
	7- end-user efficiency	5
	8- online updates	5
	9- complex processing	3
	10- reusability	4
	11- installation ease	3
	12- operations ease	5
	13- Multiple sites	3
	14- facilitated change	5
	TDI	51

→ a bit high!
Please re-check.

6.8 Total Adjusted Function Calls

IM/VAF= TDI*0.01+0.65	1.16
ATPF= UTPF*IM	87

6.9 Line of Code

To calculate the number of code lines we chose Python as the language to use and had a unit size of 40 which we multiplied by ATFP. This is the average unit size for both.

Python 100

$\text{Loc} = \text{ATFP} * \text{Language unit size}$	87	3480
--	----	------

6.10 Estimate the Efforts

Using COCOMO I model

a	b	c	MM=a*KDSIb	Tdev = 2.5*MMc
2.4	1.05	0.38		
				~6 Months

estimated team size?

other methods?

Chapter 7

Conclusion

:)

