Contents

[//Todo 1](#_Toc149513814)

[Use cases and Activity Diagrams 1](#_Toc149513815)

[How to ensure quality 1](#_Toc149513816)

[Introduction 2](#_Toc149513817)

[Overall Description 2](#_Toc149513818)

[Requirements Modelling 3](#_Toc149513819)

[Personas (what do our users want to do) 3](#_Toc149513820)

[Scenarios (how can we discover this) 3](#_Toc149513821)

[Use cases and Activity Diagrams 3](#_Toc149513822)

[Requirements 4](#_Toc149513823)

[Student self-reporting 4](#_Toc149513824)

[How to ensure quality (sequence and state diagrams) (CI and testing) 5](#_Toc149513825)

[Design 5](#_Toc149513826)

[Class modelling 5](#_Toc149513827)

[Sequence and State Diagrams 5](#_Toc149513828)

[Data storage 6](#_Toc149513829)

[Implementation 6](#_Toc149513830)

[Testing 6](#_Toc149513831)

[Test cases 6](#_Toc149513832)

[Test evaluation 6](#_Toc149513833)

[Conclusion 6](#_Toc149513834)

[What worked 6](#_Toc149513835)

[What did not 7](#_Toc149513836)

# //Todo

## Use cases and Activity Diagrams

## How to ensure quality

Class modelling

## Sequence and State Diagrams

## Implementation

## Testing

### Test cases

### Test evaluation

## Conclusion

## 

# Introduction

(to the project you are building – not the coursework)

Improving the operation of a personal supervisor system is a primary focus within the university’s computer science department, and for that matter all the departments within the university. I’ve been tasked with creating a digital system that assists students and staff in tracking engagement and providing support when necessary.

# Overall Description

(of the problem area and the problem)

The stakeholders in this project are:

* The student
* The personal supervisor
* The senior tutor
* (Could be more found through later analysis)

This system should enable those involved in the system (Stakeholders), to carry out the following use cases:

* Students should have the ability to self-report their well-being and progress at predefined time intervals.
* Personal supervisors should be able to assess the status of all their assigned students.
* Personal supervisors should be able to schedule meetings with students.
* Students should have the option to schedule meetings with their personal supervisors.
* The Senior Tutor should have access to view the status of all students and monitor how personal supervisors are engaging with the students.

# Requirements Modelling

## Personas (what do our users want to do)

* A student should be able to log a message, on how they are feeling/progressing at fixed time intervals
* The personal supervisor should be able to view their students’ messages
* Student should be able to book a meeting with the personal supervisor
* Senior tutor should be able to see the messages of all students and how personal supervisors are responding to them
* (Could be more found through later analysis)

## Scenarios (how can we discover this)

Student is using the system to log a message at the start of every week

Personal supervisor is using the system to look at and respond to the messages sent by the student

Student is falling behind with coursework, wants to book a meeting with personal supervisor to get back on track

Senior tutor is using system to view messages between personal supervisor and student

Use cases and Activity Diagrams (scenarios + diagrams) (how can we module the interactions out software will need to allow to deliver value to our users) (do after the project has been implemented)

Requirements Test Plans and Traceability Matrix

### Student self-reporting

Test that a student can log a report their progress at fixed time intervals – check a log has been completed

Verify the system records and stores the self-reported data accurately in the database (storage system) – data entered is the same as the data entered

### Personal supervisor reviewing student status

Test whether a personal supervisor can access and view the status of all their assigned students. – ensure the status of all students are visible

Verify that the displayed student information is accurate and up to date

### Personal supervisor booking a meeting

Test the personal supervisor’s ability to book a meeting with a student – test whether the student and the personal supervisor can see the meeting

Ensure that the meeting is correctly scheduled and added to the systems calendar

### Student Booking a meeting with personal supervisor

Test that a student can successfully request a meeting with their personal supervisor - test whether the student and the personal supervisor can see the meeting

Ensure the PS is notified, and the meeting is scheduled

### Senior tutor Monitoring student status

Test that an ST can access a dashboard displaying the status of all students and how personal supervisor is interacting with them

Verify that the data is accurate and provides insights into students -PS interactions

### Data persistence

Close the program and reopen it to test whether changes made (e.g., meetings/progress reports) are still present, indicating data persistence

### Invalid input handling

Attempt to input invalid data, such as non-numeric characters in progress reports or scheduling a meeting int the past.

Ensure that the system provides appropriate error messages, and prevents the input of invalid data

## How to ensure quality (sequence and state diagrams) (CI and testing)

# Design

Class modelling (and possibly object modelling) (do after the project has been implemented)

## Sequence and State Diagrams (do after the project has been implemented)

## Data storage

The application will require some form of data storage, so if changes are made and the program is closed, those changes should be persistent (the changes will be saved, so are still visible after the program is ran again).

I will be using a JSON file. This is because JSON files allow you to represent structured data, making it suitable for storing information about students, personal supervisors, meetings, and other related data in a well-organized manner. JSON files are also human readable which makes it useful for debugging, this is especially useful for development and troubleshooting.

Furthermore, JSON is a widely accepted data format for data exchange between different systems and services. If this program ever needs to integrate with other software or external services, using JSON simplifies this process. Also, JSON is a lightweight format that doesn’t add much overhead to the application size. This is important as storage on university computers is limited and slow. Finally, JSON files are text based, making them easy to manage with version control systems like GIT.

# Implementation

Implementation aspects (Brief documentation of implementation aspects)

Automated test plan (Evidence of automated test plan)

Automated build pipeline (Evidence of automated build pipeline)

Evidence of finished project (Could include screen shots/ link to video walk through)

# Testing

## Test cases

## Test evaluation

# Conclusion

## What worked

## What did not