|  |  |
| --- | --- |
|  | **2020** |
|  |  |



|  |
| --- |
| **[Future marker]** |
| **Team Members:**   * Karim Hisham Mansour * Mohamed Essam Abdelaziz * Martin Fawzy * Anas Hassan * Haitham Hashim   **Project Supervisor: Dr Essam Eliwa** |

Future Academy - Higher Future Institute for Specialized Technological Studies

School of Computer Science

Project report submitted to the Future Academy  
for the degree of Bachelor of Science

June 2020

**[Optional Project Logo]**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**[Optional Students pictures]**

**Acknowledgement**

**Abstract**

**Table of Contents**

[Chapter1 Introduction and Background 1](#_Toc4274305)

[Introduction 2](#_Toc4274306)

[Problem definition 2](#_Toc4274307)

[What is the importance of this problem? 2](#_Toc4274308)

[What are the current solutions? 3](#_Toc4274309)

[How will this solution solve the problem? What is new? 3](#_Toc4274310)

[Scope 4](#_Toc4274311)

[SummaryChapter 2 Analysis and Design 5](#_Toc4274312)

[Introduction 7](#_Toc4274313)

[User and System Requirements 7](#_Toc4274314)

[Functional requirements 7](#_Toc4274315)

[Non – functional requirements 7](#_Toc4274316)

[Stack holders 7](#_Toc4274317)

[System Design 7](#_Toc4274318)

[Block Diagram 7](#_Toc4274319)

[Use Cases 7](#_Toc4274320)

[Class Diagram 8](#_Toc4274321)

[Design Patterns 8](#_Toc4274322)

[Sequence Diagrams 8](#_Toc4274323)

[Database Design 8](#_Toc4274324)

[Used Technologies and tools 8](#_Toc4274325)

[Summary 8](#_Toc4274326)

[Chapter 3 Deliverables and Evaluation 9](#_Toc4274327)

[Introduction 10](#_Toc4274328)

[User Manual 10](#_Toc4274329)

[Testing 10](#_Toc4274330)

[Evaluation (User experiment) 10](#_Toc4274331)

[Summary 10](#_Toc4274332)

[Chapter 4 Discussion and Conclusion 11](#_Toc4274333)

[Introduction 12](#_Toc4274334)

[Main Findings 12](#_Toc4274335)

[Why is this project important 12](#_Toc4274336)

[Practical Implementations 12](#_Toc4274337)

[Future Recommendation 12](#_Toc4274338)

[Conclusion Summary 13](#_Toc4274339)

[References 14](#_Toc4274340)

**Table of Figures**

[Figure 1: example for adding photos 3](#_Toc508085361)

[Figure 2: software development life cycle 5](#_Toc508085362)

[Figure 3 12](#_Toc508085363)

# Chapter1 Introduction and Background

**Main points**

* [give a brief abstract of your chapter here]

## Introduction

Computer Science degrees are very popular study programs worldwide. A core topic in computer science is to learn programming skills. Programming is also related to several fields of technology, hence many university students -in various programs- study it. Novice programming students require to practice as much as possible to enhance their problem solving and programming skills. A key tool for instructors is setting assignment for students to encourage them to practice more. However, to gain the most required result from such assignments, quick and detailed feedback is required.

## Problem definition

The main problem with the traditional manual assignments marking is that the instructor usually need to spend a lot of time and efforts to check and mark every student's code and to write a report for every student on each assignment to give feedback for the submission that inform him that his submission was good enough or not, and advise students about the mistakes that they did and what points they have missed. Instructors also need to set a clear grading policy and give students a breakdown of the grading criteria [1]. The time required to achieve this would usually mean late feedback delivery to students, it may take days and sometime weeks for a student to know his/her marks. furthermore, the accuracy of the marking or writing feedback may decrease as the number of students increase, in addition to that some students may plagiaries code from the internet or from their friends [2], and it is hard to check plagiarism for each student manually. In addition, student can submit the assignment only once, without being able to submit it again after the feedback he receives with his grade.

### What is the importance of this problem?

By 2020 all the Educational sectors are working hard to improve themselves by using new tools and technologies or improving the old ones [3], Today we can see the importance of following up with the students as fast as possible by monitoring their level, as the instructors can evaluate them by giving them assignment and marking them, but if we are going to talk about the coding and computing we have to admit that evaluating the students by the instructor is not easy and it requires more efforts and time to check the coding level and give them a feedback for every single task, hence providing instructors with easy to use automated code assessment tool that can evaluate assignment in no time with minimum setup, and giving instant feedback to student and send a report for the instructor about each student and what grade they take and what points they miss [4], This will help student to enhance their programming skills vastly through learning from their mistakes, it will also enable instructors to set more assignments without the worry of the time they need to mark it. In addition, it will save a lot of time for the instructors and professors so that they can be more creative with their teaching with the students.

### What are the current solutions?

For many decades automatic assessment has been applied in many forms even before personal computers existed. Automatic assessment had already been suggested by Hollingsworth in 1960 [5].

In many programming courses at tertiary institutions the use of automated assessment has been proved useful through the use of systems like:

-The CourseMaster Automated Assessment System (2001) [6]

-The BOSS Online Submission and Assessment System (2005) [7]

-Individualized exercises for self-assessment of programming knowledge: An evaluation of Quiz PACK (2005) [8]

-ALOHA - A Grading Tool for Semi-Automatic Assessment of Mass Programming Courses (2006) [9]

-Easy Accept: a tool to easily create, run and drive development with automated acceptance tests (2006) [10]

-PASS - Programming Assignment assessment System (2006) [11]

-Automatic marking with Sakai (2008) [12]

-Web-CAT: Automatically Grading Programming Assignments (2008) [13]

-Programming Task Packages: Peach Exchange Format (2008) [14]

-PROGTEST: An Environment for the Submission and Evaluation of Programming Assignments based on Testing Activities (2011) [15]

### A System to Grade Computer Programming Skills using Machine Learning (2014) [16]How will this solution solve the problem? What is new?

This project introduces Future Marker (Automated Task Assessment Cloud based System). It will reduce the effort and time consumed by instructors when they assess submitted tasks from students. It will assess tasks in one of two ways. The first one is by using logical technique which begin with compile the code to check for runtime errors, then evaluating typographic layout which is checking for the layout and indentation, besides the identifiers’ name and length and the written comments, then checking code efficiency by running multi dynamic test cases. The second one is by utilizing machine learning and natural language processing to make it even more easy to instructors to setup assignments.

Future Marker will solve many problems for students. It will allow students to submit assignments a number of times customized by the instructor to improve their grades and learn from their mistakes by getting instant feedback from the system. Future Marker will also check and report any plagiarized work through comparing submitted students’ assignments to each other. It will also compare submissions with previous years submissions for similar assignments and automatically search for and report similar solutions online.

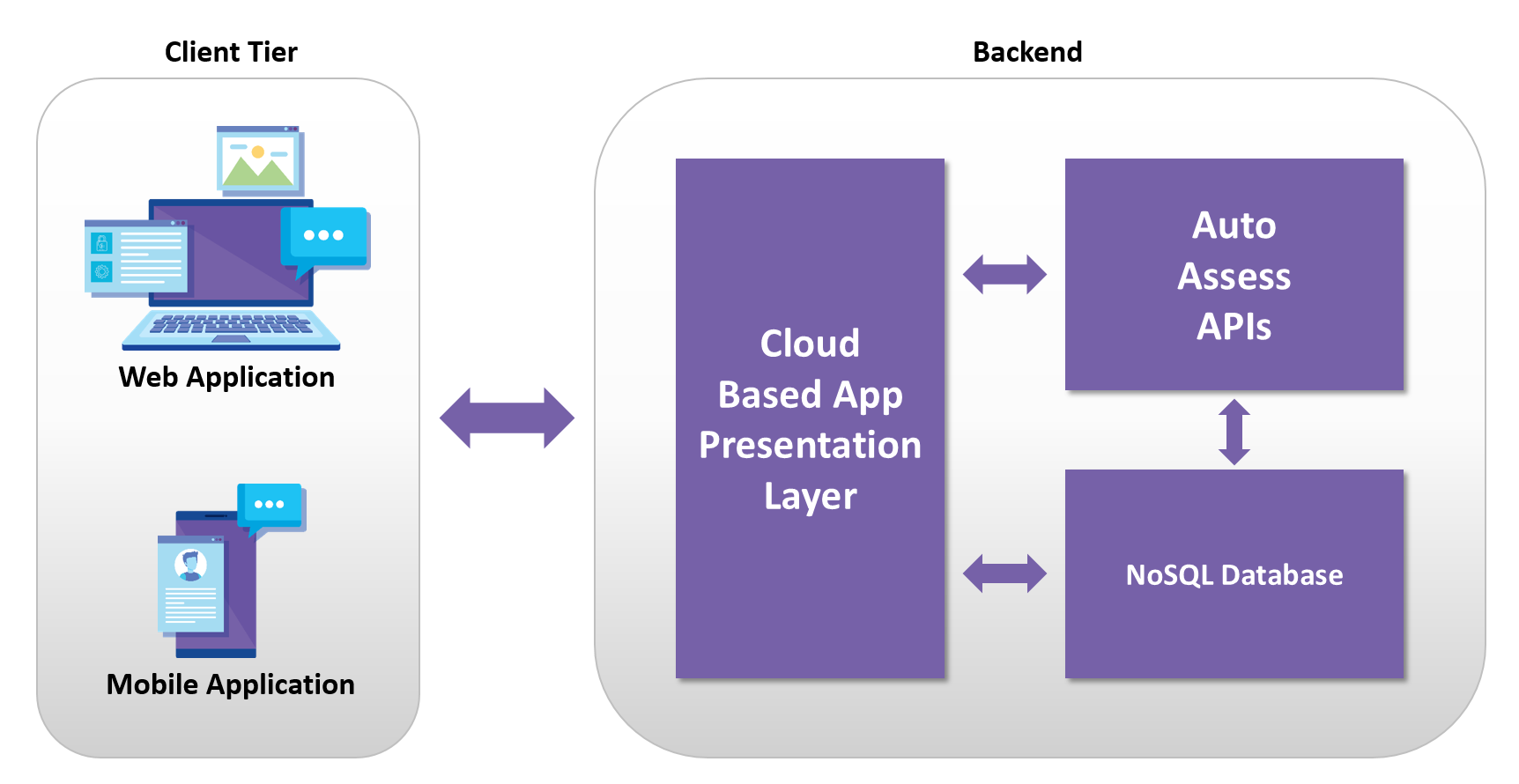
Future Marker aims to utilize the recent technological advancement in both cloud computing and artificial intelligence to provide a smart, accessible, and easy to use system for programming instructors worldwide.

**Project Description**

Future Marker is a 3-tier application architecture that consists of a presentation tier,

an application tier and a data tier.

* Presentation tier - This tier consists of web application and mobile application. The web application will be programmed using HTML5, cascading style sheets (CSS) and JavaScript. The mobile application will be programmed using flutter framework to provide mobile app for android and IOS
* Application tier - Contains the business logic of the system. It will be programmed using PHP, java and python, also it will be hosted on distributed servers in the cloud. Application tier communicates with the other tiers through application program interface (API) calls.
* Data tier – In this tier we will use Firebase as a database for the system because it’s a cloud-hosted NoSQL database that lets you store and sync between your users in real time.

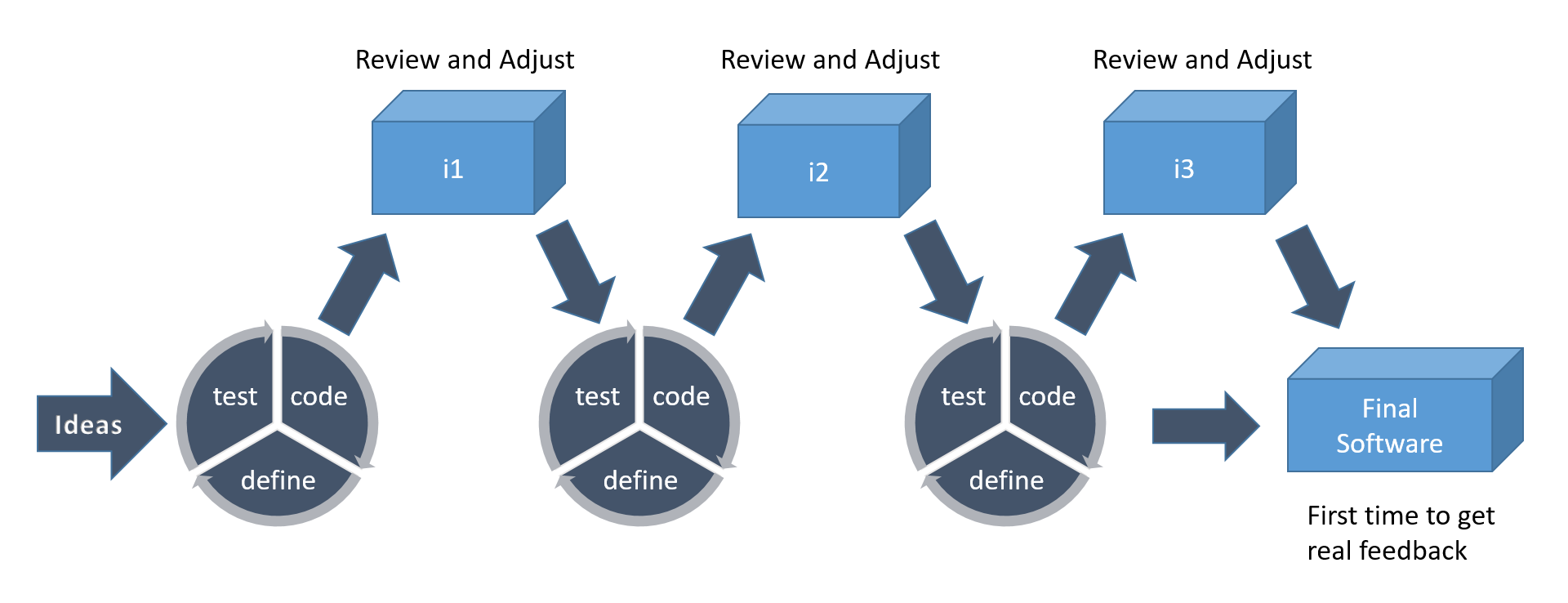


## Scope

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor.

**Methodology (Proposed Approach)**

We will use agile methodology with incremental plan because it is easier to change the process to reflect changing customer requirements and the process works well when not all requirements are known.



We will use scrum method because that focuses on managing iterative development, users see on-time delivery of increments and gain feedback that will help us to improve our system. We will have the flexibility to change anything while working on the system.

## Quality assurance will consist of several stages:

* Review of requirements
* Test planning / writing test cases
* Unit testing
* Integration testing
* System testing
* Performance testing
* Security testing
* Cross-browser testing / cross-platform testing
* Updating test cases
* Regression testing

**Deliverables**

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor.

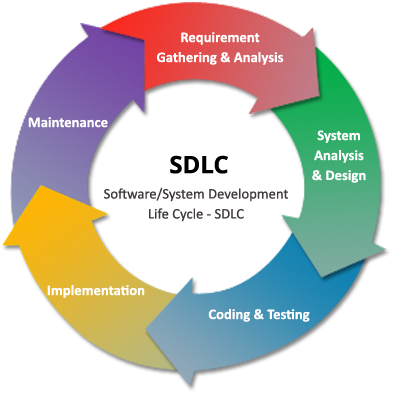


Figure : software development life cycle

# SummaryChapter 2 Analysis and Design

**Main points**

* [give a brief abstract of your chapter here]

## Introduction

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor.

## User and System Requirements

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor.

### Functional requirements

### Non – functional requirements

## Stack holders

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor.

## System Design

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor.

### Block Diagram

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

### 

### Use Case Diagram

### Class Diagram

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

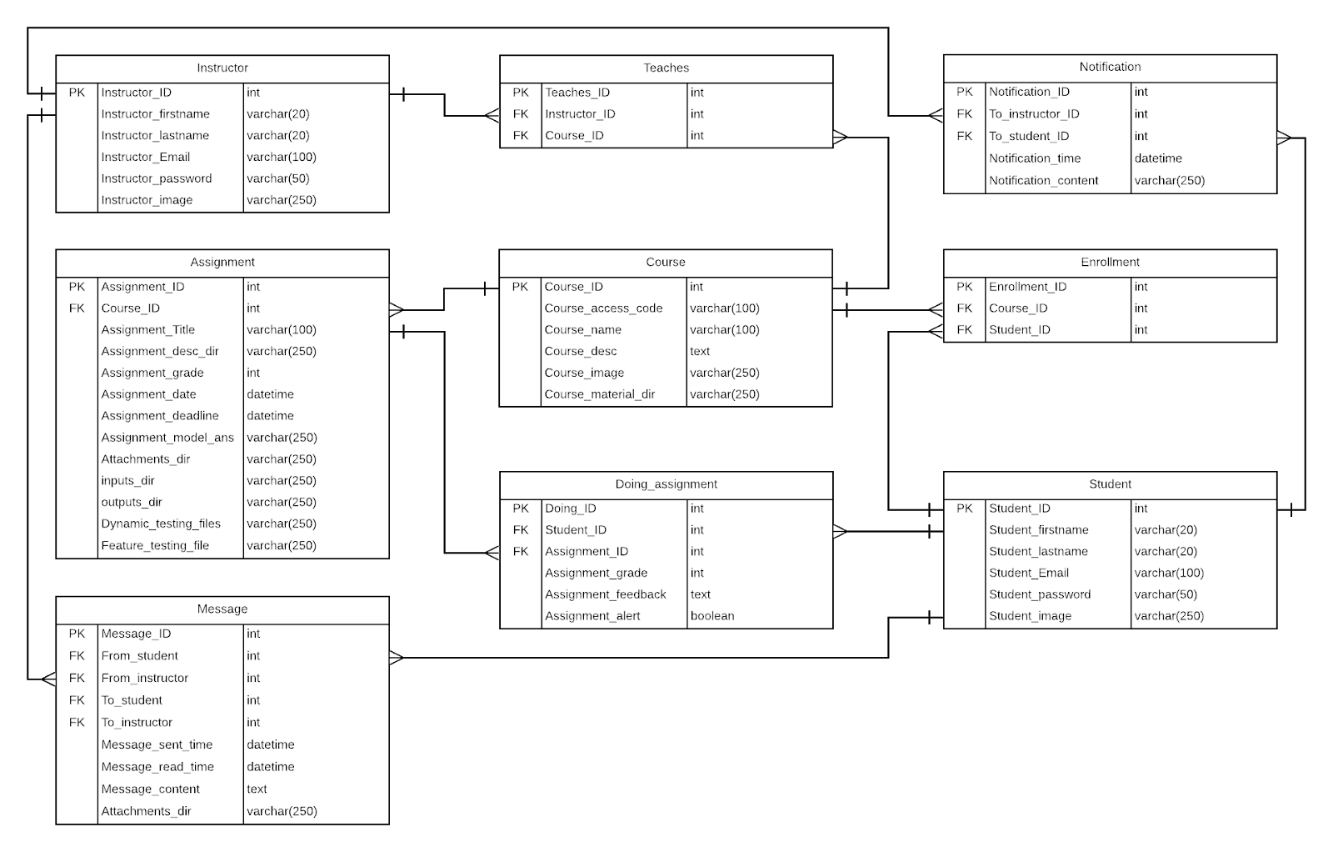
### Design Patterns

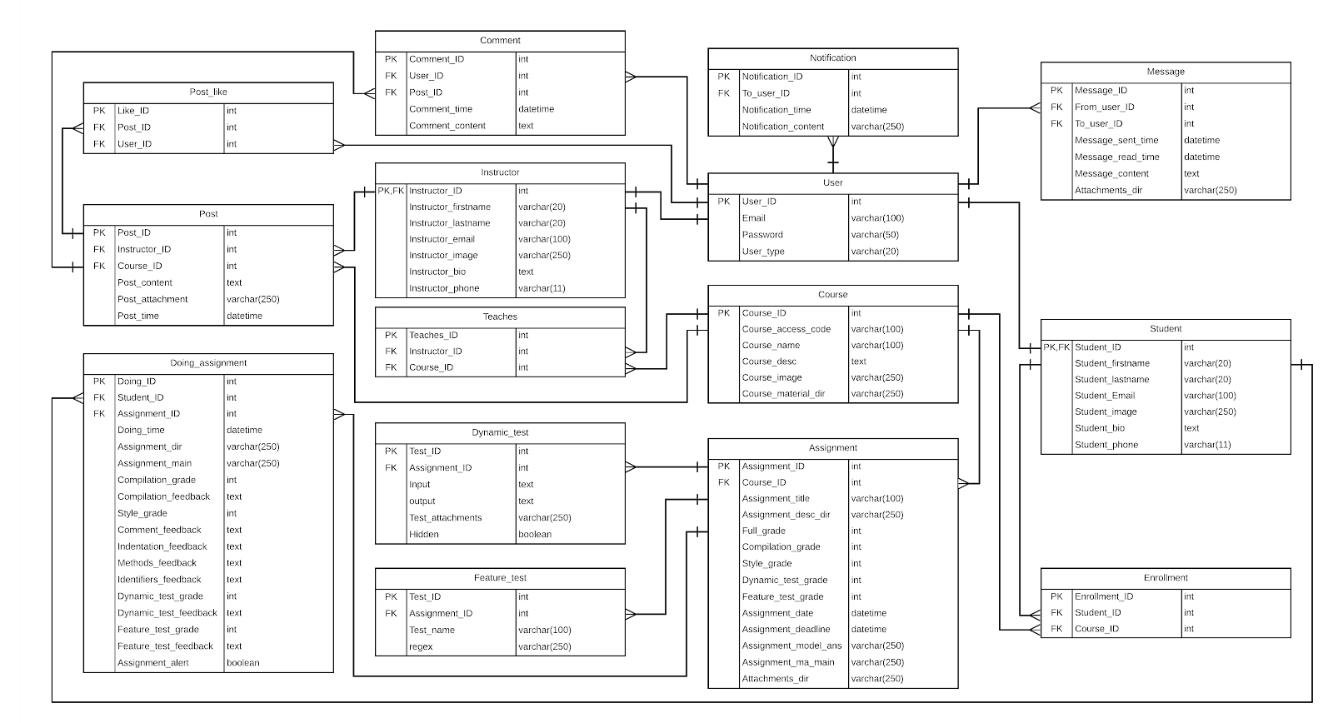
Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

### Sequence Diagrams

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

### Database Design (ERD)





## Storage Structure

## https://lh5.googleusercontent.com/ceD-NyYPJyLg_GD1X_JvFTlg-H710zdwOd_ogjGBO_o_q1gCRK9zrxmQub9_W11uD4f0Iz1ghrb9y3hF2ceNps3MNYaMyiUe-wEXnzipxHT3fiqJhswR3jHEtAYZT9oTsOwz9VhwXMU

## Used Technologies and tools

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

Summary

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

# Chapter 3 Deliverables and Evaluation

**Main points**

* [give a brief abstract of your chapter here]

## Introduction

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

## User Manual

## Testing

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

## Evaluation (User experiment)

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

Summary

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

# Chapter 4 Discussion and Conclusion

**Main points**

* [give a brief abstract of your chapter here]

## Introduction

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.



Figure

## Main Findings

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

### Why is this project important

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

### Practical Implementations

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

## Future Recommendation

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

## Conclusion Summary

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

# References

|  |  |
| --- | --- |
| [1] | D. Gooblar, “news,” 4 January 2017. [Online]. Available: https://chroniclevitae.com/news/1656-how-can-we-minimize-grade-challenges. |
| [2] | L. T. Culwin F, “ Plagiarism issues for higher education,” *Emerald Inshight,* pp. 36-41, 2001. |
| [3] | T. A. Khan, “buyer-blog/how-technology-improves-education,” 28 march 2019. [Online]. Available: https://www.trustradius.com/buyer-blog/how-technology-improves-education. |
| [4] | V. A. Shashank Srikant, “A System to Grade Computer Programming Skills using Machine Learning,” 2014. |
| [5] | HOLLINGSWORTH, “The morphology of intersexes in Drosophila subobscura,” *The Journal of experimental zoology,* vol. 143, 1960. |
| [6] | E. a. H. C. a. S. P. a. T. A. Foxley, “The CourseMaster Automated Assessment System - a next generation Ceilidh,” 2001. |
| [7] | N. G. R. B. Mike Joy, “The boss online submission and assessment system,” 2005. |
| [8] | S. S. Peter Brusilovsky, “Individualized exercises for self-assessment of programming knowledge: An evaluation of QuizPACK,” September 2005. |
| [9] | T. R. Tuukka Ahoniemi, “aloha title : a grading tool for semi-automatic assessment of mass programming courses,” February 2006. |
| [10] | O. L. A. N. W. C. Jacques Philippe Sauvé, “EasyAccept: a tool to easily create, run and drive development with automated acceptance tests,” May 2006. |
| [11] | T.-L. W. Fu Lee Wang, “Designing Programming Exercises with Computer Assisted Instruction,” August 2008. |
| [12] | H. Suleman, “Automatic marking with Sakai,” 2008. |
| [13] | M. A. P.-Q. Stephen H. Edwards, “Web-CAT: automatically grading programming assignments,” September 2008. |
| [14] | T. Verhoeff, “Programming Task Packages: Peach Exchange Format,” 2008. |
| [15] | J. C. M. E. F. B. Draylson Micael de Souza, “ProgTest: An environment for the submission and evaluation of programming assignments based on testing activities,” 2011. |
| [16] | S. a. A. V. Srikant, “A system to grade computer programming skills using machine learning,” 2014. |