# Introduction

With increasing dependency of humans on technology, many aspects of how we operate has been influenced by it. One of the many aspects which fall under the influenced category is E-Learning. As of this era, learning via electronic media is starting to change the conventional standards of education. Assimilating new ideas and knowledge has never been resourceful to this degree and increasing.

# Justification of project

As a student, we invest plenty of time in electronic devices like smart phones and laptop. With E-learning, utilizing modern technologies to one’s educational benefit results in better academic performance along with personal growth. Makes it relatively easier for students with unfortunate disabilities to assimilate knowledge.

# Background of Project

As per the statistics suggests, most of us spend our time in front of our screen. Exploiting this fact, why not make this time fruitful by investing in something as productive as learning new skills and techniques. The intention behind the project is simply delivering reliable knowledge materials who are interested in learning programming technologies with furthermore packed features.

# Problem Statement

Conventional learning methodologies is turning out to be limited. Spirited students yearning to learn programming either join classes far from their residential area or consult relevant books. For students with unfortunate disabilities, it might be problematic to follow these conventional means.

In order to overcome this problem, this project is designed so that every passionate student is able to assimilate the necessary programming concepts via electronic devices they possess.

# Description of the project

The following project is about providing genuine concepts related to programming. The contents within the project scope provides a wide variety of syllabus categorized according to variants of programming languages used in the market.

The tutorial platform not only provides a learning environment, but also an environment where students are able to share their knowledge.

# Features

* Provision of tracking progress so that the students are able to learn at their own pace.
* The admin can create and manage programming courses at any given time so that the available courses are up to date.
* Achievements and awards for devoted students to arouse a feeling of accomplishment.
* Gamification features for making the learning process more engaging with friendly competition.
* Social learning via student groups and community.
* Provides mobile learning features which is responsive across various platform.

# Overview of the project

The tutorials provide on a particular topic delivers necessary knowledge about the programming language and how it is implemented. The application also provides group discussions and question sections where student themselves communicate with one another to share what they have learnt.

Accordingly, achievements and awards for completing a tutorial is provided to students which evokes a sense of accomplishment. This not only makes the course competitive and motivating, but also positively aids one’s mental health.

# Scope of the project

The following encompasses the scope of interactive tutorial platform.

* Delivery of course contents to varied audience. This includes reaching out to learners with physical disabilities, learners from rural areas etc.
* Availability of rich contents to audience that was unreachable earlier.
* Teachers in rural village areas can be deliver course contents using a single electronic device.

# Limitations

The following explains some of the limitations within the system.

* Students must have uninterrupted internet connection in order to access the tutorial contents.
* No recovery options for students willing to recover their profile once it has been deleted.

# Aims of the project

* Make the concepts and study materials of programming and its discipline available for passionate learners.
* Enable learners from remote areas, full-time workers, women with small children etcetera to access programming knowledge at their will via electronic devices.

# Objectives

* Provide login and registration of learner profile for student.
* To be able to edit and delete learner profile at will.
* Provide proper navigations on the front-end section so that students can focus more on the content rather than navigating what they are looking for.
* Including comment sections after each course where students are able to leave questions or share something more about the respective topic.
* Provide achievements and awards after completion of each course.
* Provision of creating social groups and community.
* Include a forum section where queries and questions are processed and discussed among learners.

# Development Methodologies

Before plunging into the deep processes and mechanisms of the system, we first consult certain development methodology which is suitable for our project. Development methodologies are utilized in order to make the entire development process structured and provide a momentum to the project.

In case of this project, we consult the Waterfall Model also known as the traditional methodology.]

# Description of Methodology

The Waterfall methodology is considered a classic approach and is widely used during software development process. The entire approach complements the methodology name itself. As in waterfall, the development process divided into multiple phases. Each phase occurs after the completion of the phase prior to it. The following figure depicts how Waterfall model functions.

Figure : Waterfall methodology

Feasibility

Maintenance

Analysis

Design

Implementation

Testing

* **Feasibility**

The potential requirements are identified and defined in this phase. In our case, the specification of what the learner wants.

* **Analysis**

We produce proper models and business logic based on the system specifications identified in the previous phase.

* **Design**

A system design is generated where necessary software and hardware specifications are mentioned which is further utilized to define the system architecture.

* **Coding/Implementation**

The system design is further implemented to produce a viable source code which utilizes the models and logic generated in the previous phase.

* **Testing**

The source code is tested in units and merged into a functional system after successful testing of each unit.

* **Maintenance**

In case of unwanted issues in the client side, patches are used to overcome the encountered issues.

## Advantages of Waterfall model in project scenario

* Since the project itself is intermediate, it is easy to manage.
* Provides clear idea and picture of requirements.
* Ensures effective completion of a phase before proceeding to another.
* The milestones can be identified easily.
* Documentation of the project is well-defined and well-produced.

# Design Pattern

Out of numerous available design patters, Model View Controller (MVC) pattern seems most viable and effective for the project.

## Model View Controller (MVC)

Via MVC, the intended application is divided into three logical parts, model, view and controller. The following figure depicts how each component communicates with one another.

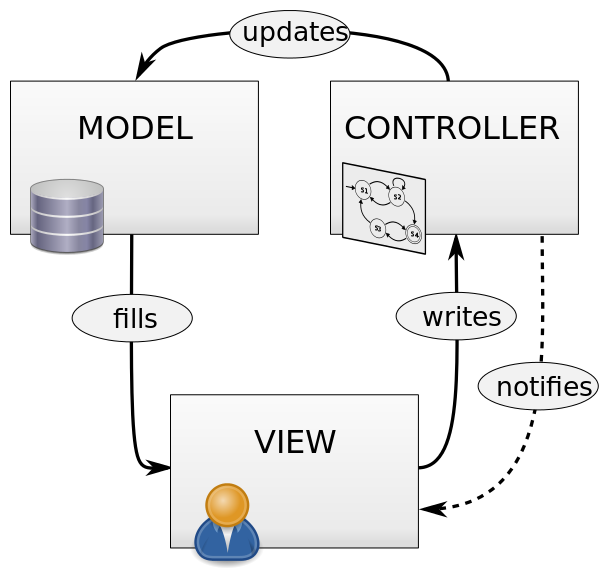


Figure : MVC flow

* **Model**

Simply put, model deals with the data of the application. It communicates directly with the database and treats the data as objects of a model for specified entity.

* **View**

View components directly interact with the users and deals with the UI of the application. It basically renders output processed by controller.

* **Controller**

The controller deals with the logic and request as an intermediate between view component and model.

# Architecture

I decided to build the entire project on a three-tier architecture which is comprised of three tiers Presentation tier, Application tier and Data tier. The MVC pattern discussed prior is implemented in Presentation layer of the architecture. The following depicts how the three tiers communicate with one another.



Figure : Three-tier architecture flow

# Tools

**IDE** – IntelliJ IDEA 2018.3.5

**Programming language** –

**Frameworks used** – Bootstrap

**Programming paradigm** – Object Oriented Programming

**Database** – MySQL

**Server solution stack** – XAMPP

**Modelling** **tool** – Visual Paradigm

# Project Planning

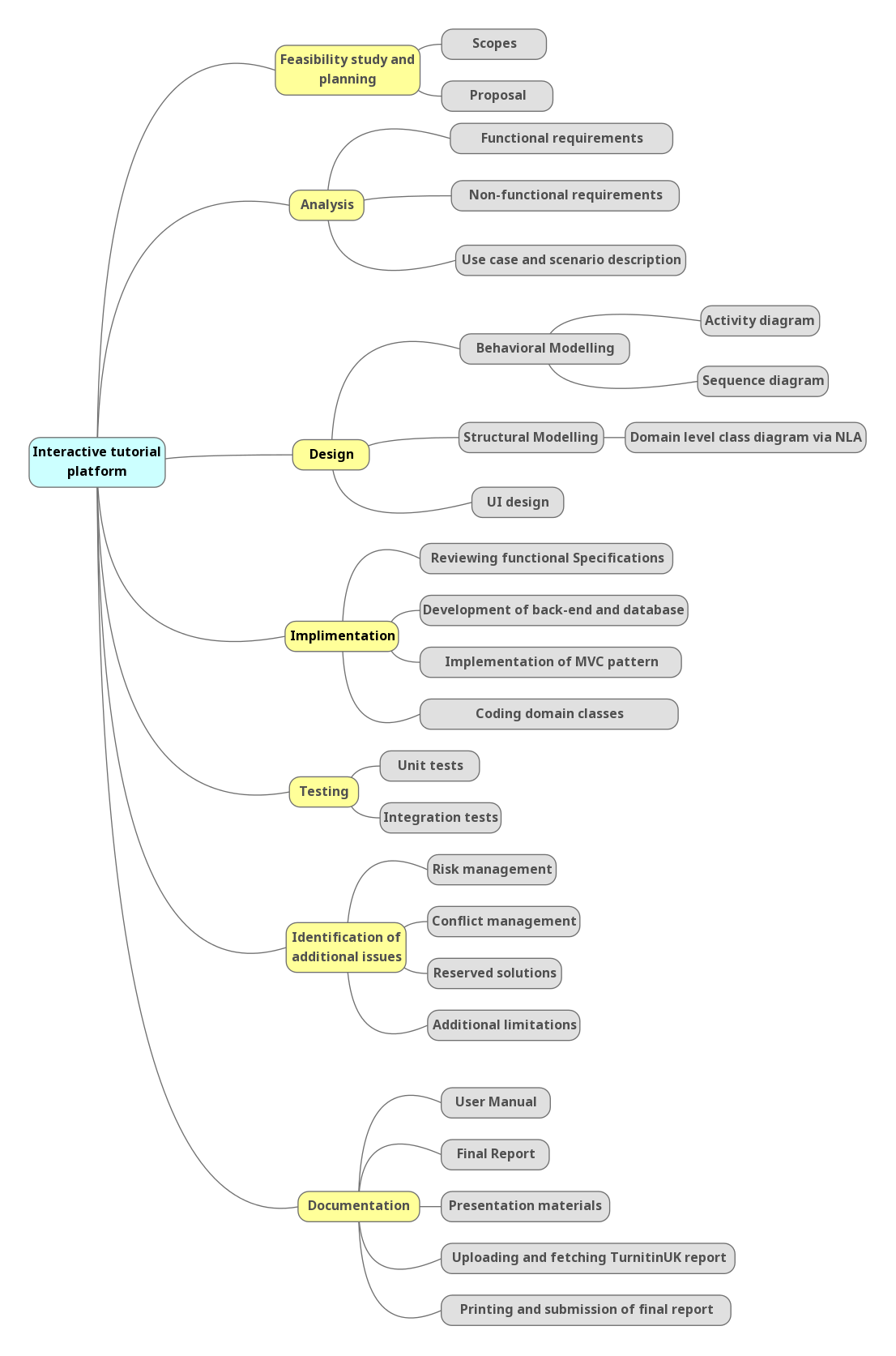


Figure : Project planning

# About Work Breakdown Structure (WBS)

WBS is an efficient way of dividing and conquering large scale projects. It comprises of multiple hierarchical structures which is relatively more manageable and easier to work with. The following WBS table depicts the estimations and approximations of time required for completion of relative tasks.

|  |  |  |
| --- | --- | --- |
| WBS | Task Description | Estimated time in days |
| 0 | **Interactive tutorial platform** | **109** |
| 1 | **Feasibility study and Planning** | **15** |
| 1.1 | Scoping | 5 |
| 1.2 | Proposal Writing | 10 |
| 2 | **Analysis** | **29** |
| 2.1 | Achieve functional requirements | 11 |
| 2.2 | Achieve non-functional requirements | 11 |
| 2.3 | Use case and scenario description | 7 |
| 3 | **Design** | **26** |
| 3.1 | Behavioral Modelling | 12 |
| 3.1.1 | Sequence diagram | 7 |
| 3.1.2 | Activity diagram | 5 |
| 3.2 | Structural Modelling | 9 |
| 3.2.1 | Domain level class diagram via NLA | 6 |
| 3.2.2 | ER diagram | 3 |
| 3.3 | UI design | 5 |
| 4 | **Implementation** | **21** |
| 4.1 | Reviewing functional specifications | 5 |
| 4.2 | Development of backend and database | 7 |
| 4.3 | Implementation of MVC pattern | 5 |
| 4.4 | Coding domain classes | 4 |
| 5 | **Testing** | **7** |
| 5.1 | Perform unit testing | 4 |
| 5.2 | Perform Integration testing | 3 |
| 6 | **Documentation** | **11** |
| 6.1 | User Manual | 2 |
| 6.2 | Final Report | 3 |
| 6.3 | Presentation materials | 3 |
| 6.4 | Upload and fetch final report to TurnitinUK | 1 |
| 6.5 | Print and submit hardcopy | 2 |

# Milestones

Figure : Project milestones

# Project Schedule

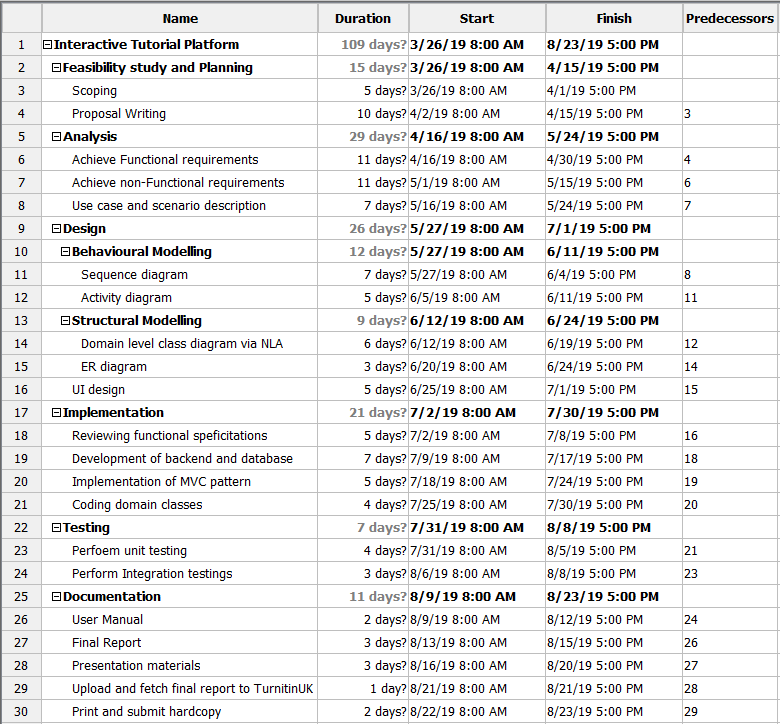


Figure : Project Schedule

# Gantt Chart

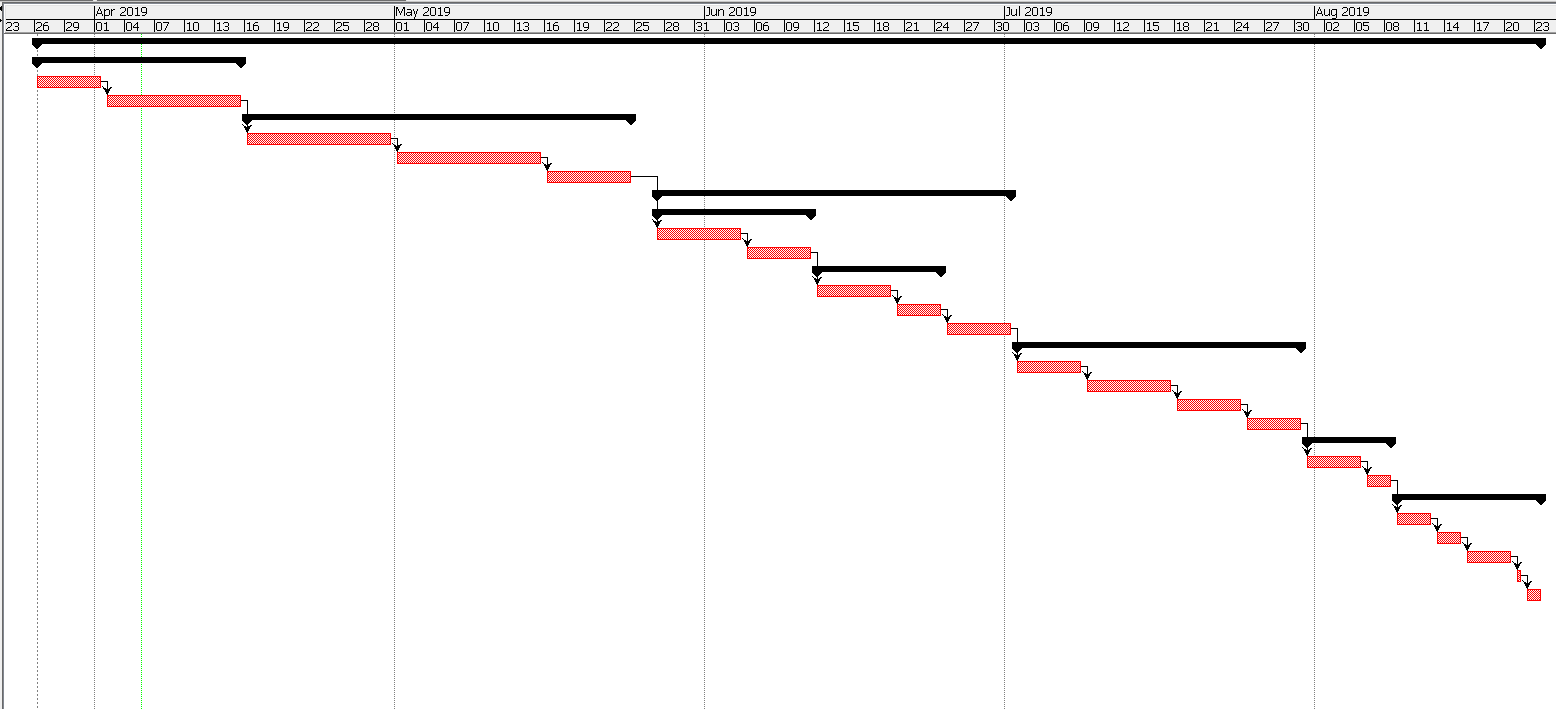


Figure : Gantt chart

# Risk Management

No matter how robust our system is, we somewhat manage to pack some risk when we build and refurbish a system. The unforeseen casualties somewhere in the future gets triggered when we ignore risks or leave it uncalculated.

When we address potential risks in order to understand and analyze them, we refer to it as Risk Management. Keeping various aspects of the system in mind, we calculate the risk and manage them such that the system remains healthy and safe. The steps involved in risk management are as mentioned below.

* Risk identification
* Access impact of risk
* Lessen critical risk
* Risk control

|  |  |
| --- | --- |
| **Likelihood** | **Value** |
| Low | 1 |
| Medium | 2 |
| High | 3 |

|  |  |
| --- | --- |
| **Consequences** | **Value** |
| Very Low | 1 |
| Low | 2 |
| Medium | 3 |
| High | 4 |
| Very high | 5 |

With strong reference to the tables mentioned above, we calculate the impact of a specific risk with respect to the formula:

**Impact = Likelihood \* Consequences**

In case of my project, the following risk management matrix is presented which follows certain Risk Management standards.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Risk Type** | **Description** | **Likelihood** | **Consequence** | **Impact** | **Action** |
| Non-Technical | Natural Disaster | 1 | 5 | 5 |  |
| Deadline Overruns | 2 | 4 | 8 |  |
| Unforeseen growth of project scope | 1 | 5 | 5 |  |
| Gold plating – including features irrelevant to project scope | 2 | 4 | 8 |  |
| Technical | Encountering unwanted bugs and glitches | 3 | 4 | 12 |  |
| Hardware failure | 2 | 5 | 10 |  |
| Complex algorithms resulting memory leakage and CPU overhead | 2 | 5 | 10 |  |
| Hard-disk and backup failure | 1 | 5 | 5 |  |

# Configuration management