**Chapter 1**

**1.1 Background**

Every semester, there are countless students who would like to decide how to proceed in their respective courses and also customize it. So, it is extremely important that the student give the required information like the intake, number of semesters, waivers, prerequisites etc., The present software does not allow for any modifications in the course structure and the student cannot alter it according to their needs.

**1.4 Project Scope**

This project is completely focused on the task of generating a map of all the courses the student will take as required by the computer science department which includes all the compulsory courses, general courses and waivers. The project will not generate a map for a non-BSCS course and is strictly restricted to the computer science department. The student must have a seamless experience with the developed web pages and the time taken for the overhead must be contained to a little. Such an approach would result in giving the student a proper understanding of the department and its conditions before getting enrolled in it.

**2.4** **Benefits of software proposed**

The advantage of the suggested software is that it creates a more focused, easy-to-use interface with additional fields to assist students in getting a comprehensive course map for the duration of the BSCS Program. Students enrolled in the BSCS program will have their general needs met by this software, which can also be customized to meet the needs of other UMSL campus programs.

**Chapter 3**

**3.1 Introduction**

For this project, Agile development will be selected over all other software development approaches. Throughout the project's construction, the team's approach to each development phase and communication style were determined by this process.

**3.2 Software Engineering Methodology**

The agile model is an iterative software development process that guarantees the production of a functional software deliverable at the end of each iteration. Every iteration in agile development comprises incremental features and processes, and the final iteration produces a software product with all the features and components the client has specified as requirements.

‘

**3.3 Justification for Methodology**

Since each member of our team is working on a separate module, the Agile technique is appropriate and practical for our project. Each iteration allows a team member to work on their own module and, when finished, connect it with other modules to create a deliverable. Once the dependencies are assessed, tested, and released between those independent modules with each iteration, the necessary adjustments can be applied to our project. In addition, a lot of flaws and other relevant problems could go unnoticed for a while. These can be addressed in later versions to provide a stable product at last.

**3.3.1 UML Methodology**

UML stands for Unified Modeling Language, and it was the methodology used for system design, modeling, and analysis. This approach was chosen due to the language's popularity and the elegance with which diagrams are constructed in it. A clear and useful model of the system might be produced by employing UML to construct use-case, sequence, activity, architecture, and class diagrams.

**3.3.2 Methodology Prototype**

Simple modules for login, User information and course selection made up the system prototype, to which further modules were eventually added. With the help of this prototype, the development team was able to test deployment, establish a GitHub repository, and gain a better understanding of Flask's features.

**Chapter 4**

**Activity Diagram**

**Use Case for editing courses**

**Use Case**: Edit Courses

**Primary actor**: UMSL bachelor’s degree student

**Goal in context**: To edit BSCS courses in a generated map for adding, moving, removing and printing the information of the courses.

**Preconditions**: Course map creator has been programmed to recognize required courses and realistic duration of the specified degree program. Semester to begin map planning is specified.

**Trigger**: Student selects a course from the generated map and performs an action

**Scenario**: 1. Student picks courses

2. Student edits and moves courses to the semester of his liking.

3. Student checks the validity of his course map plan.

4. Student gets the course map.

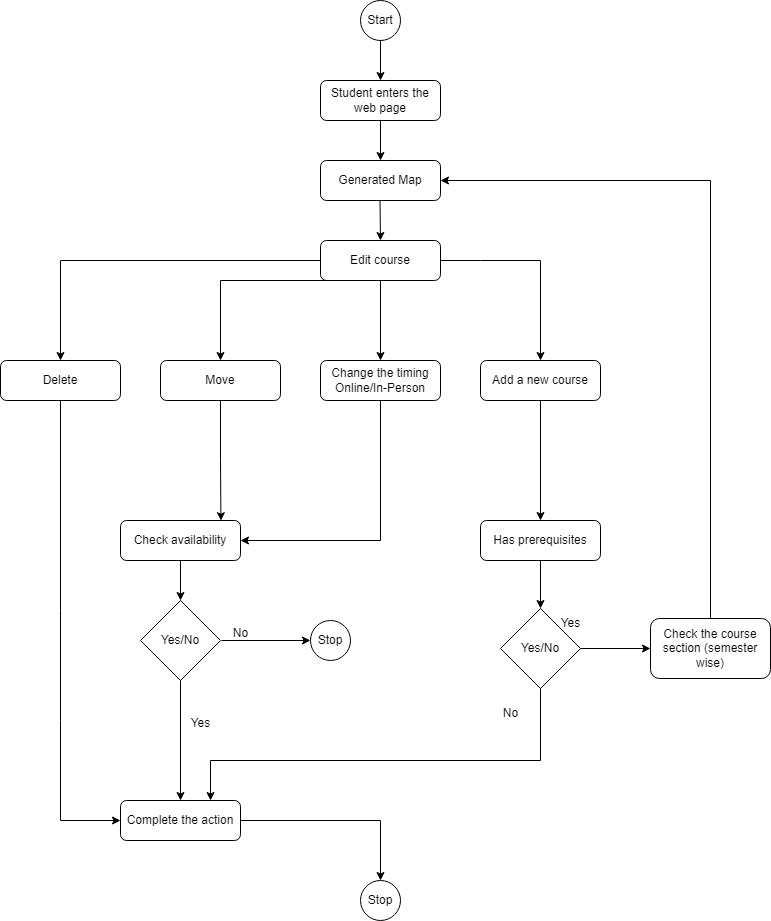
**Exceptions**: None

**Priority**: Essential must be implemented

**When available**: On deployment

**Secondary actors**: Filesystem administrator

**Activity Diagram**:



**Sequence Diagram**:

