7.2 Artifacts

|  |  |
| --- | --- |
| 1 | Member Organization |
|  | Team members subdivided themselves into groups of programmers and documenters according to skill set. |

|  |  |
| --- | --- |
| 2 | Initial Discussion |
|  | The initial discussion involves the gathering of ideas and information around the project. Initial software requirements and designs were proposed. Team members shared their key strengths in order to participate effectively in the process. A brainstorming of the structural and functional requirements of the scheduler lead to the establishment of a foundation for the software development process. |

|  |  |
| --- | --- |
| 3 | Deliverable 0 |
|  | Deliverable 0 consisted of all the elements that were discussed in the initial discussion. A description for the project was provided, along with the domain model diagram and a description of the areas within the domain model. An initial list of the team members and the tasks they were interested in accomplishing regarding the application was also drawn up. |

|  |  |
| --- | --- |
| 4 | Scheduling of Tasks |
|  | This artifact involved assigning specific tasks to each member of the team regarding the project. Meeting times for each subgroup working on a certain task were also arranged and set up, as well as dates as to when each task should be expected to be completed. |

|  |  |
| --- | --- |
| 5 | Resource Evaluation |
|  | Establishing everyone's technical capacities and the amount of time each individual can spend working on the project. Collected every technical resource available to the team including software tools and computers. |

|  |  |
| --- | --- |
| 6 | Framework Configuration |
|  | Downloading and installing laravel on each programmer's machine. Importing the various libraries needed to make run the application through the use of composer. Importing the database. |

|  |  |
| --- | --- |
| 7 | Collaboration Tools |
|  | Setting up every team member with all collaboration tools. Creating a slack team to set up meetings, to discuss and to communicate useful information. Creating a Google Drive for all team members to contribute documents, in order to work on collaboratively. Creating a GitHub organization for all teams members to access the repository which includes documents and codes. This allows to keep track of any revisions and issues that occur. |

|  |  |
| --- | --- |
| 8 | Domain Model Design |
|  | Creating the domain model diagram to document the key concepts and terminology of the system. It consists of domain level objects, their attributes and associations. |

|  |  |
| --- | --- |
| 9 | Course Database |
|  | Adding to the course database the course titles, professors, time slots, requisites, descriptions, credits and faculty of each course required for the software engineering degree. Inputting the course lecture, tutorial, and lab time slots. |

|  |  |
| --- | --- |
| 10 | Prototype |
|  | Creating a live prototype to allow team members to input course information to the database. The prototype will demonstrate the integration of laravel and angular.js frameworks for our application. |

|  |  |
| --- | --- |
| 11 | Use Case Tables & Diagrams |
|  | The use case tables describe in detail every possible user (actor) interaction with the system. The use case diagrams visualize these interactions, grouped by actors. This artifact is important for the identification of functionalities of the system. |

|  |  |
| --- | --- |
| 12 | Constraints Identification |
|  | This artifact involves analyzing and classifying any constraints that the team might have. It includes design, hardware, performance, security, or external constraints. The identified constraints are then described in detail. |

|  |  |
| --- | --- |
| 13 | Scoping |
|  | The scoping artifact describes the functionalities that are removed from the final system design due to limited time or resources. The functionalities may include features, goals, or requirements and qualities of the system. For each scoped out functionality, a clear reason is given. |

|  |  |
| --- | --- |
| 14 | Initial Front End Planning & Design |
|  | Establishing design patterns to follow when creating the user interface. Choosing a color palette. Drawing mock-ups. |

|  |  |
| --- | --- |
| 15 | Estimation & Risk Planning |
|  | Carefully identifying risks that the project can have so team members can take pre-caution to avoid those risks. Estimating the total cost of the project after carefully estimating the cost of production of each artifact. Also estimating the time for each task and scheduling accordingly. |

|  |  |
| --- | --- |
| 16 | Deliverable 1 Documentation |
|  | Deliverable 1 consisted of the initial project description and domain model which was provided in deliverable 0. In addition to that, after several team meetings we defined the goals of our project in the document along with all the constraints, some of which were scoped down later on. Human and technical resources were discussed and tasks were split accordingly. An early overview of our solution and plan along with a prototype was provided in this document. |

|  |  |
| --- | --- |
| 17 | Architecture Design & Diagrams |
|  | An updated and detailed version of the architectural design. It includes a description of the reasons behind our design, the changes we made from the previous design in Deliverable 1, and all the components (such as function calls and description of parameters) included in our new design. This will be presented in the form of UML diagrams for our complete description of the system design. |

|  |  |
| --- | --- |
| 18 | Frontend Implementation |
|  | Development of the frontend of our system, the part where the user can interact with. |

|  |  |
| --- | --- |
| 19 | Backend Implementation |
|  | Development of the backend of our system, which includes the server, the application and the database. |

|  |  |
| --- | --- |
| 20 | UML Class Diagrams |
|  | Create Unified Modelling Language (UML) class diagrams for all classes of program. Responsible person who meet with developing team. |

|  |  |
| --- | --- |
| 21 | Dynamic Design Scenarios |
|  | A full dynamic design of use cases including system sequence designs, operational contracts and sequence diagrams. |

|  |  |
| --- | --- |
| 22 | Deliverable 2 Documentation |
|  | Deliverable 2 consists of a much more structured and detailed design for our system, including better UML class diagrams. It also includes an updated, and more a detailed version of the architectural design from the previous one. Dynamic design diagrams are also well put together. This deliverable will help the programmers give a more organised overview of the system, which they can report on its rapid prototypes and examine the effects on estimates, risks and scopes. |

|  |  |
| --- | --- |
| 23 | Test Planning & Testing |
|  | Meet with the programmers and discuss the program’s possible weaknesses, compile a list. Meet with the testing team and plan out use cases specific to weakness discussed with developers; also plan other cases covering all other aspects the testing team can derive. Compile use cases, test them on program and record success, failures, and comments. |

|  |  |
| --- | --- |
| 24 | Improvements |
|  | Implement risk management for all problems found during testing; treat problems found as identified risks. Implement risk analysis (prioritizing), risk planning (mitigation, avoidance, and reprogramming), and create document for risk monitoring to be part of system administrator’s user manual. |

|  |  |
| --- | --- |
| 25 | User Manual |
|  | Create User manual for end user and system administrator. |

|  |  |
| --- | --- |
| 26 | Final Cost Estimate |
|  | A final listing of all components of all phases of the project, including the hours cost per person for each of the components of each phase. |

|  |  |
| --- | --- |
| 27 | Deliverable 3 Documentation |
|  | Deliverable 3 consists of documenting the testing process, including unit, requirement, stress and security testing. Includes the user manual, and final cost estimate. |

|  |  |
| --- | --- |
| 28 | Project Documentation & Presentation |
|  | This includes the final and complete documentation of the project. A presentation is given to demonstrate the fully functional application. |

1. Member Organization (tasks)
2. Initial Discussion (looking at requirements etc)
3. Deliverable 0

Deliverable 0 consisted of all the elements that were discussed in the initial discussion. A description for the project was provided, along with the domain model diagram and a description of the areas within the domain model. An initial list of the team members and the tasks they were interested in accomplishing regarding the application was also drawn up.

1. Scheduling of Tasks

This artefact involved assigning specific tasks to each member of the team regarding the project. Meeting times for each subgroup working on a certain task were also arranged and set up, as well as dates as to when each task should be expected to be completed.

1. Resource Evaluation (tech & human)
2. Framework Config
3. Collaboration Tools (slack, github, drive)
4. Domain Model Design
5. Course Database
6. Prototype
7. Use Case Tables & Diagrams
8. Constraints Identification
9. Scoping
10. Initial Front End Planning & Design
11. Estimation & Risk Planning
12. Deliverable 1 Documentation
13. Architecture Design & Diagrams
14. Frontend Implementation
15. Backend Implementation
16. UML Class Diagrams
17. Dynamic Design Scenarios
18. Deliverable 2 Documentation
19. Test Planning & Testing
20. Improvements
21. User Manual
22. Final Cost Estimate
23. Deliverable 3 Documentation
24. Project Documentation & Presentation