**CIS 422**

**Initial Project Plan (IPP) for UNAMED\_CALENDAR**

**CIS422**

**Team Number 6**

**499ms**

**Initial Project Plan**

**Document**

**Version :1 Date: (1/15/2019)**

**Team Member Assignments**

|  |  |  |
| --- | --- | --- |
| **Role** | **Team Member** | **Main Responsibilities** |
| Project Manager | Noah Palmer | Designate and assign tasks to all team member. Ensures all milestones will be met. Adjusts requirements as needed. |
| Minute-Takers | Primary: Zachary Bower  Secondary: Chase Craig | Maintains an accurate written record of team meetings. |
| System Architecture and Design | Primary: Noah Palmer  Secondary: Ben Yin | Maintaining adherence to documentation and project schedule |
| GUI | Primary: Chase Craig  Secondary: Zachary Bower | Creating all view controllers for user interaction and information display |
| Codebase Developer | Primary: Refael Yehuda  Secondary: Ben Yin | Ensuring code base follows the information flow layed out in the documentation |
| Issue/Bug Tracking | Primary: Zachary Bower  Secondary: Chase Craig | Track all known bugs and issues with the repository update when solved |
| Quality Control | Primary: Noah Palmer  Secondary: Zachary Bower | Ensure codebase follows code standards laid out in CSD. Ensure documentation is both grammatically and logically correct |
| User Documentation | Primary: Ben Yin  Secondary: Refael Yehuda | Create user help documentation to assist with common tasks |
| Technical Documentation | Primary: Zachary Bower  Secondary: Chase Craig | Create documentation for all objects and classes in the project to aid developers in production |

**Milestones**

|  |  |
| --- | --- |
| Date | Goals |
| January 12 2019 | Assign members documentation  Agree upon project specifications including (Language, GUI, Native/Web based) |
| January 15 2019 | Rough draft of SRS/SDS completed  Assign development tasks  Training for database and GUI  Second meeting notes: |
| January 17 2019 | Initial SRS Document complete  Initial SDS Document complete  Initial IPP Document complete |
| January 18 2019 | Meeting with Hornof  Basic GUI completed (limited functionality)  Bare Bones Database  Third meeting notes: |
| January 22 2019 | MVP for backend (program logic) |
| January 24 2019 | MVP for frontend (user logic) |
| January 26 2019 | Complete unit testing for communication between front and backend systems |
| January 29 2019 | Technical documentation complete for all existing code |
| February 1 2019 | Begin final testing and debugging  All basic requirements are tested and verified.  Finalize non essential features to add |
| February 3 2019 | All unit tests for non essential features complete |
| February 5 2019 | Final Application |

**Team Communication and Workflow**

Team 499ms meets twice a week, the first weekly meeting is on Tuesdays from 15:00 - 16:00 , the second is on Fridays from 10:00 - 11:00. Each meeting begins with a progress report. Each team member will spend around three minutes explaining the progress they have made on assigned tasks. After each team member has discussed their progress, the meeting moves toward ensuring the milestones above are met. Team members are encouraged to speak with each other regarding any help needed to complete a task. Depending on the date of the meeting and which milestone is being sought, team members will begin a code review process.

Team 499ms provides a communication server through Discord all members have joined. Members may use this communication server to discuss matters related to any issues or needs of the project.

**Adherence to Coding Standards**

All team members submitting code to UNAMED\_CALENDAR will adhere to coding style standards described in the code style sheet ( [https://docs.google.com/document/d/1KL8Q\_9lhUDCnlcPMZtaPfaDTiaKeyBGpLgd7bN3cE5s/](https://docs.google.com/document/d/1KL8Q_9lhUDCnlcPMZtaPfaDTiaKeyBGpLgd7bN3cE5s/edit) ). All code submitted to the code base will be original unless cited.

**Documentation Review Procedure**

During the first meeting the team discussed the documentation process from inception to final product. The team members decided upon a peer review process that follows this flow.

1. The initial version of a document is created
2. The Editor reviews the submitted document for errors/inconsistencies
3. The author of the paper makes these changes
4. The Senior Editor reviews the submitted document for errors/inconsistencies
5. The author of the paper makes these changes
6. The final version of the document is submitted

**Code Review Procedure**

Code reviews will be assigned to all team members, and be due 24 hours after assignment. Code reviews are to ensure all code created is of the highest quality possible, and to ensure the coding style document is followed.

**Code Review Process**

Before code is merged into the final repository a team members code must go through a code review process.

1. Team members are assigned feedback from a rotating member of the team.
2. The member of the team assigned to review the piece of code ensures the following items:
   1. Adherence to Coding Style Standard
   2. The code is easy to comprehend and documented accordingly
   3. The code follows the design and functionality listed in the SRS/SDS
   4. The code is optimized accordingly

(3) The team member who submitted the code is given the reviewed code within 24 hours

(4) The team member submits the code to the test repository

(5) The code is reviewed by either the primary or secondary lead for the code base

1. GUI code is reviewed by the GUI team
2. Backend/codebase code is reviewed by the codebase team

(6) The code is merged with test repository and ready for testing

**Work Progress and Management**

All progress on the project will be tracked through Github. The main project page will have multiple branches for different features being worked on at any time. The two main branches will be a the master (final project) and the test branch.

Developers will be working on a branch other than the main two at any time. When the branch has gone through the code review process and is ready for testing, the branch will merge with the test branch.

When the test branch receives an update, the quality control team will begin testing. If the changes pass testing, the test branch is merged with the master.

**Work Progress Rationale**

The above workflow was chosen due to its simplicity. Different models were proposed and discussed during team meetings, but due to the team size and the scope of the project separating the project into multiple branches and merging was found as the best choice.

The use of two main branches, one for final changes and one for test changes allows for team members to synchronize issues, updates and work to be done. The use of separate branches for new features then merging allows the project to maintain a modular design while avoiding conflicts within the repository.

**Issue and Test Reporting**

Any issues and bugs found during testing will be reported on Github using the issues tab on Github. Issues may also be discussed in person during meetings, or online via the team communication server, but all issues must be explicitly written in Github.

When an issue or bug is reported on Github, the issue tracking team begins working on the test branch to solve the reported issue. When the issue has been corrected the issue tracking team must perform all previous tests to ensure all previous work has not been altered. If the update passes all tests the issue is closed and the master branch is updated.

**Risk Assessment and Management**

Multiple risks exist during all stages of development, the following risks have been accessed by the development team and strategies to overcome these risks have been discussed.

1. A developer being unfamiliar with a programming language or dependency used
   1. To lessen this risk, team members will share resources and help one another both in person at meetings, or in the communication server every member of the development team has access to
2. An uneven distribution of work
   1. To lessen this risk, team members are encouraged to ask for help as needed to finish a project. The workload and milestones have been created with this in mind and are modular to allow for easy manipulation of team members assigned task if needed
3. Deadlines not being met
   1. To lessen this risk milestones will act as deadlines for all team members

**Meeting Notes and Attendance**

Each meeting two team members will take attendance and give an overview of the topics discussed in the meeting. These notes are located in the link below.

Meeting notes

[January 12, 2019](https://docs.google.com/spreadsheets/d/1ZaPJSS47yXwEG5gK-AAlZHn4-U91EKBHmvZ77K_yCb4/edit#gid=380732291)