# CIS 422

# Initial Project Plan (IPP) for YACC

# CIS 422

Team Number 6

# 499ms

Zachary Bower, Chase Craig, Refael Yehuda, Noah Palmer, Benjamin Yin

Initial Project Plan

Version 1.2 Date: (02/04/2019)

Table of Contents

# 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<br>Secondary: Chase Craig | Maintains an accurate written record of team meetings. |
| System Architecture and Design | Primary: Noah Palmer<br>Secondary: Ben Yin | Maintaining adherence to documentation and project schedule |
| GUI | Primary: Chase Craig<br>Secondary: Zachary Bower | Creating all view controllers for user interaction and information display |
| Codebase Developer | Primary: Zachary Bower<br>Secondary: Chase Craig | Ensuring code base follows the information flow layed out in the documentation |
| Issue/Bug Tracking | Primary: Refael Yehuda<br>Secondary: Chase Craig | Track all known bugs and issues with the repository update when solved |
| Quality Control | Primary: Noah Palmer<br>Secondary: Zachary Bower | Ensure documentation is both grammatically and logically correct |
| User Documentation | Primary: Ben Yin<br>Secondary: Refael Yehuda | Create user help documentation to assist with common tasks |
| Technical Documentation | Primary: Zachary Bower<br>Secondary: Chase Craig | Create documentation for all objects and classes in the project to aid developers in production |

# Work Breakdown and Milestones:

| Milestone | Date | Contributors and Time Spent | Status |
|---|---|---|---|
| Initial Planning and Design | 1/12 | Everyone - 2.5 Hours | Complete |
| Initial SRS, SDS, and IPP | 1/13 | Everyone - 20 Hours | Complete |
| Assign Developer’s Tasks | 1/15 | Zach (Database)<br>Chase (GUI)<br>Noah (Event Handling)<br>Refael (GUI)<br>Ben (Error Handling) | Complete |
| Rough Draft of SRS, SDS, and IPP | 1/17 | Zach (SRS/IPP) - 4 Hours<br>Chase (SRS/SDS) - 3 Hours<br>Noah (SDS) - 5 Hours<br>Refael (SDS) - 4 Hours<br>Ben (SDS) - 3 Hours | Complete |
| Meeting with Hornof | 1/18 | Zach - 1 Hour<br>Chase - 1 Hour<br>Noah - 1 Hour<br>Ben - 1 Hour | Complete |
| Barebones GUI and database | 1/18 | Zach (Database) - 1 Hour<br>Chase (GUI) - 5 Hours | Complete |
| Error Handler Built | 1/20 | Ben - 2 Hours | Complete |
| MVP Backend | 1/22 | Zach - (Database) - 3 Hours<br>Chase (Code Reviews) - 4 Hours<br>Noah (Event Handling) - 4 Hours<br>Refael - 4 Hours | Complete (1/25) |
| MVP Frontend | 1/24 | Chase (Main View) - 4 Hours<br>Refael - 2 Hours | Complete (1/27) |
| Review and Mark Changes for SRS, SDS, and IPP | 1/25 | Zach (SRS/IPP) - 1 Hour<br>Chase - 2 Hours<br>Noah (SDS) - 3 Hours<br>Refael (SDS) - 3 Hours<br>Ben (SDS) - 1 Hour | Complete (1/26) |
| Begin Module Integration | 1/28 | Zach - 5 Hours | Complete (1/31) |

| | | | |
|---|---|---|---|
| | | Chase - 14 Hours<br>Noah - 1 Hour | |
| Begin User Documentation | 1/28 | Noah - 2 Hours<br>Refael - 2 Hours<br>Ben - 2 Hours | Complete (2/3) |
| Begin Developer Documentation | 1/28 | Zach - 8 Hours<br>Ben - 1 Hours | Complete (2/2) |
| SRS, SDS, IPP Update/Review | 1/29 | Zach (SRS/IPP) - 2 Hours<br>Noah (SDS) - 4 Hours<br>Refael (SDS) - 2 Hours | Complete (1/31) |
| Begin Final Product Testing and Debugging | 1/30 | Zach - 1 Hour<br>Chase - 12 Hours<br>Noah - 1 Hour<br>Ben - 1 Hour | Complete (2/3) |
| Finalize Data Format | 1/30 | Zach - 1 Hour<br>Ben - 1 Hour | Complete |
| Update Database and SDS for New Format | 1/31 | Zach - 2 Hours<br>Noah - 2 Hours | Complete (2/1) |
| Begin Data Testing and Parsing | 2/1 | Zach - 2 Hours<br>Chase - 4 Hours<br>Noah - 2 Hours | Complete (2/3) |
| Finalize Features to be Added | 2/1 | Everyone - 1 Hour | Complete |
| Meeting with Hornof | 2/1 | Everyone - .5 Hours | Complete |
| Update Documentation per Meeting Notes | 2/1 | Everyone - 4 Hours | Complete |
| User Documentation and Developer Documentation Complete | 2/3 | Zach (Developer) - 2 Hours<br>Noah (User) - 2 Hours<br>Ben (User) - 2 Hours<br>Refael - 2 Hours | Complete (2/4) |
| All Testing Complete | 2/4 | Zach (Linux) - 4 Hours<br>Chase (Windows) - 6 Hours<br>Noah (Linux) - 4 Hours<br>Refael (MacOS) - 2 Hours<br>Ben (Windows) - 2 Hours | Complete (2/4) |

# 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.

# 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 document is reviewed changes are suggested
3. The author of the paper makes these changes
4. 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:
    a. The code is modular in nature and documented accordingly
    b. The code follows the design and functionality listed in the SRS/SDS
    c. The code is optimized accordingly
3. The code is merged 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. To report a bug a developer will adhere to the following steps.

1. Open a new issue on Github
2. Add a description of the issue

After an issue/bug has been reported a member of the bug tracking team will begin work on the issue. Once corrected the code must be tested prior to finalizing the change. If new bugs are found, a new issue will be reported according to the above steps.

# 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.

   a. 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
   a. 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.
   a. To lessen this risk milestones will act as deadlines for all team members. Additionally, Trello will be a project management software implemented to track progress.