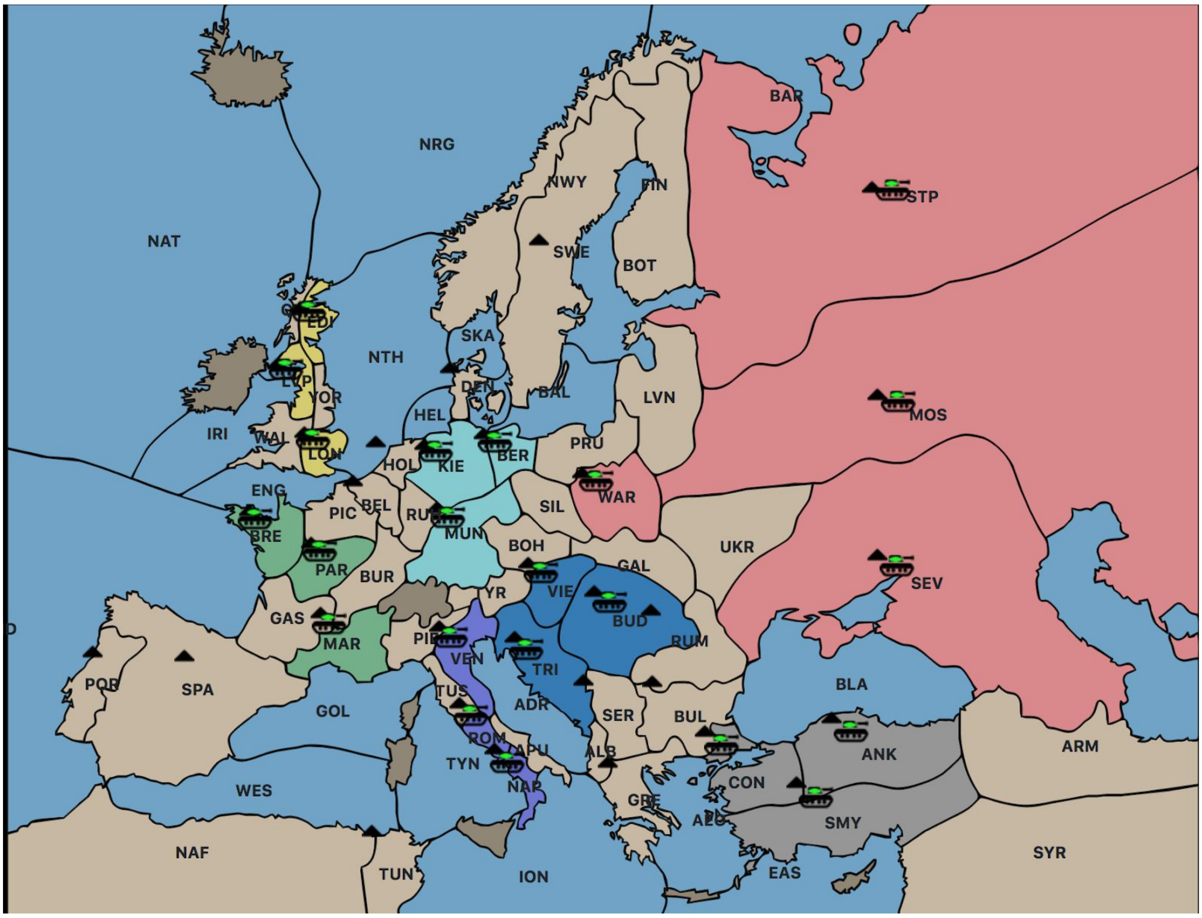
# Diplomacy

## Project Plan



**Project Development Team B:**

Ahmed Arbi<br>
Steven Duong<br>
Benson Gao<br>
Eleazar (Eli) Gomez<br>
Michael Scheid<br>
Justin Terry<br>
Vincent Tran

## Contents

# 1. Overview

## *1.1 Motivation*

The motivation for this project is to create an application that will embody the board game Diplomacy. The application should closely follow the rules of Diplomacy. The Diplomacy application will be coded on the Electron Framework. The Electron framework enables software based off it to be portable to the desktop, mobile, and web environment. Though it should be able to be played on these environments, this project will focus on making the game playable on desktop first. If time permits, the game will be tested and developed on the other environments.

## *1.2 Deadline*

The project should be finished by the end of the semester which is 09 May 2019. The documentation will be done by 7 March 2019 and a prototype should be made by 21 March 2019. The project will not cost anything other than man hours. The project will deliver an extension of an existing product as the project will be based off the board game, Diplomacy. The deadline to deliver the project is 7 May 2019.

## *1.3 Cost*

There is no monetary cost associated with this project. It will only cost the grade of the members of this project if the project is a failure. Another cost is trust of potential users of the project. If the project is a failure then a good number of potential users would no longer trust anything created by us. We would have to deliver a good number of good projects in a row to bring back those potential user's trust. Game development is very volatile so delivering a good first project would help us immensely.

## 2. Goals and Scope

### *2.1 Project Goals*

| Project Goal | Priority | Comment/Description/Reference |
|---|---|---|
| **Functional Goals:** | *1* | |
| User Login | 2 | Should let users login and store their stats |
| Interactable Map | 4 | User should be able to do their moves on the map |
| Multiplayer | 1 | Through WAN or LAN |
| User Chat | 2 | Private Chat and Public Chat |
| Should Handle up to 7 Players | 2 | Players should have no problems connecting to up to 7 people |
| Rules enforcement | 1 | Rules should be strictly enforced |
| Asynchronous Turns | 2 | Turns should be asynchronous |
| **Technological Goals:** | 2 | |
| Speed | 2 | Game should be smooth |
| Interoperability | 3 | Game should be playable on desktop, mobile, and web |
| Usability | 2 | Game should be easy to play |
| Accessibility | 10 | If time permits, enable the game to be played by blind or deaf end-users |
| **Project Goal** | **Priority** | **Comment/Description/Reference** |
| | | |
| **Quality Goals:** | *3* | |
| Moves should be smooth | *3* | No lag |
| Moves should be checked if they are valid | 3 | Rules should be enforced |
| No crashes | 2 | Game shouldn't crash |
| Game shouldn't be too big | 2 | No 40gb game |
| **Constraints:** | 1 | |
| Desktop first | 1 | The goal for this project is to allow the game to be playable on a desktop but since the Electron allows the other environments like web and mobile this can be changed. |

| | | |
|---|---|---|
| Electron Framework | 1 | The project must be made using the Electron framework |

### *2.2 Project Scope*

#### 2.2.1 Included

The project will deliver the following:

1. Documentation
    a. Vision Document
    b. Project Plan
    c. Use Cases
    d. Test Plan
    e. Program Flow Chart
    f. User Manual
2. The Game
    a. Playable on a desktop
    b. Multiplayer functionality
    c. Login functionality
    d. Chat functionality
    e. Map functionality
    f. Rule enforcement when playing the game

#### 2.2.2 Excluded

This project will exclude:

- Marketing
- Distribution
- Bug Reports
- Community Management
- Tutorial
- Random Matchmaking
- No hosting servers
- No help in deploying to a company's server

## 3. Organization

### *3.1 Organizational Boundaries and Interfaces*

The environment the project will be embedded in is the desktop environment. External stakeholders of the project is dependent on is the professor and technically owner of all the groups. The group is dependent on the results because it will result in a grade for the class. There is no really boundaries between any of the entities in the project because there are not that many layers for the purpose of this project. There is only the interaction between the group and the professor. It is possible to interact between other groups but it would not be working towards the same project.

## *3.2 Project Organization*

### 3.2.1 Project Manager

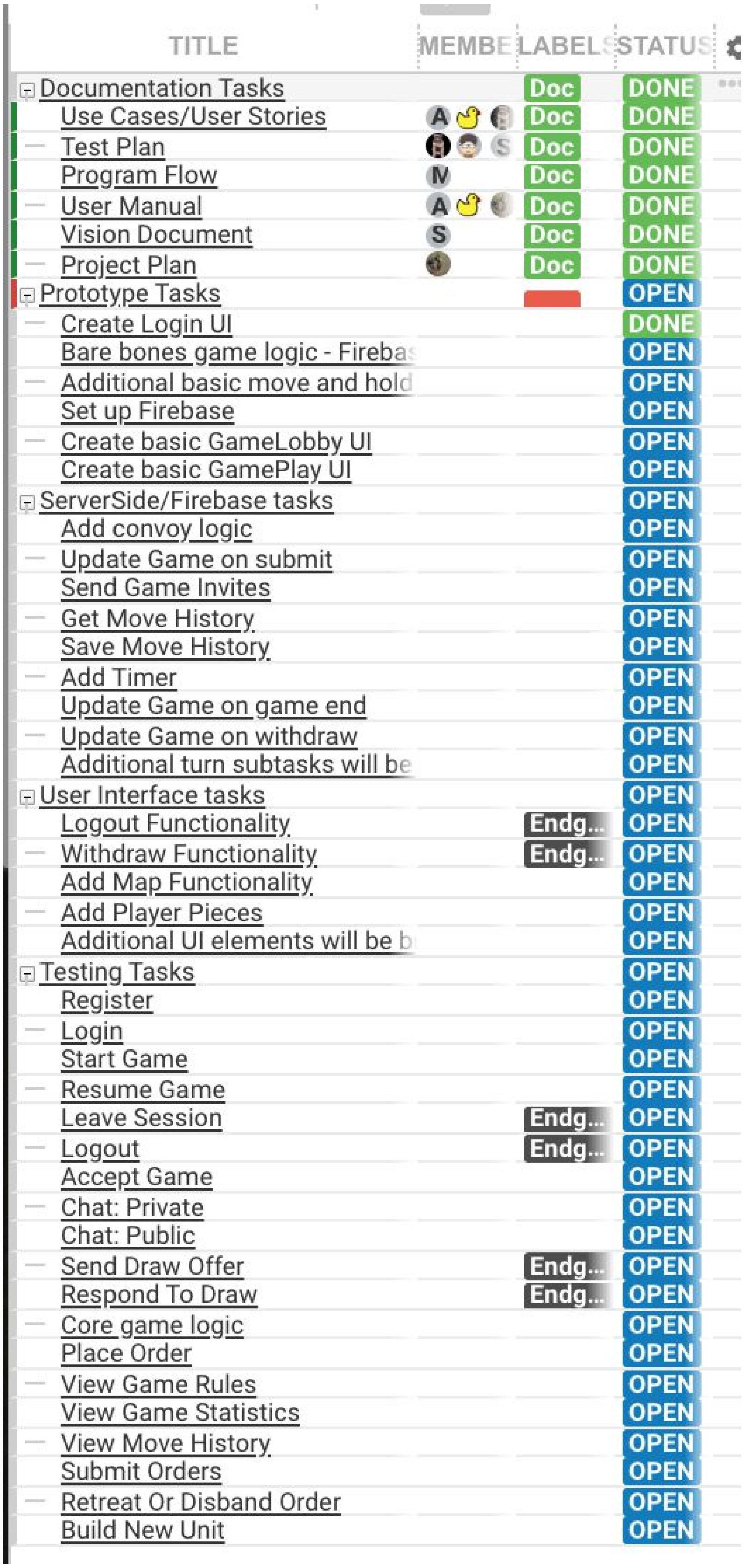
| Role | Organization: Name |
|---|---|
| Project Manager | Michael Scheid |
| Project Manager Assistant | Vincent Tran |

### 3.2.2 Project-internal Functions

| Function | Organization: Name | Concentration |
|---|---|---|
| Software Engineer | Ahmed Arbi | Database, Full Stack, Use Cases |
| Software Engineer | Benson Gao | Frontend/UX, User Manual |
| Software Engineer | Steven Duong | Frontend/UX, Vision Document |
| Software Engineer | Vincent Tran | Database, Project Manager Assistant, Project Plan, User Manual |
| Software Engineer | Michael Scheid | Project Manager |
| Test Engineer | Eli Gomez | Test Cases, Test Plan |
| Test Engineer | Justin Terry | Test Plan, Test Cases |

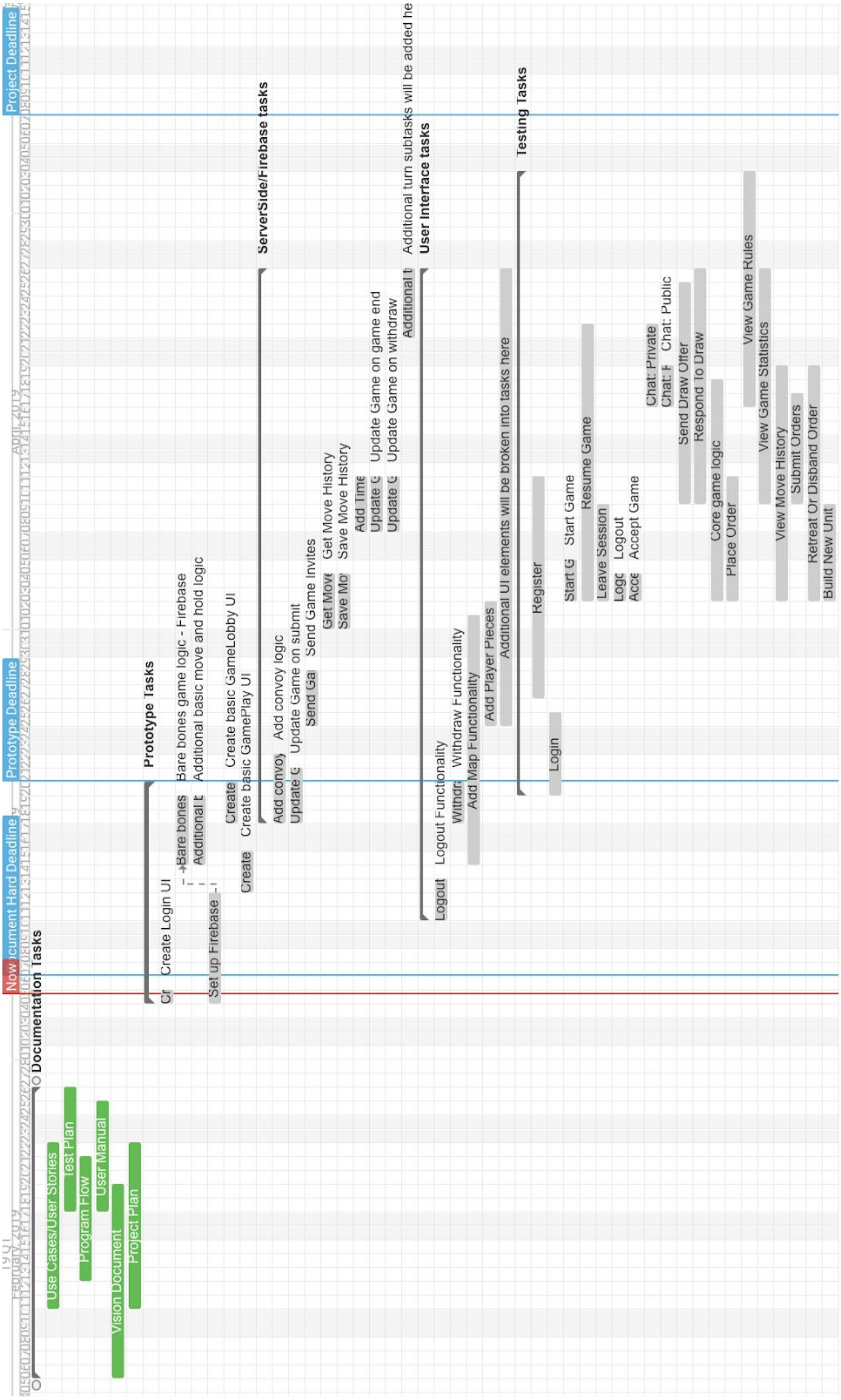
## 4. Schedule and Budget

### *4.1 Work Breakdown Structure*



| TITLE | MEMBE | LABEL | STATUS |
|---|---|---|---|
| ⊟ Documentation Tasks | | Doc | DONE |
| Use Cases/User Stories | A 🐤 | Doc | DONE |
| Test Plan | S | Doc | DONE |
| Program Flow | M | Doc | DONE |
| User Manual | A 🐤 | Doc | DONE |
| Vision Document | S | Doc | DONE |
| Project Plan | | Doc | DONE |
| ⊟ Prototype Tasks | | 🟥 | OPEN |
| Create Login UI | | | DONE |
| Bare bones game logic - Fireba | | | OPEN |
| Additional basic move and hold | | | OPEN |
| Set up Firebase | | | OPEN |
| Create basic GameLobby UI | | | OPEN |
| Create basic GamePlay UI | | | OPEN |
| ⊟ ServerSide/Firebase tasks | | | OPEN |
| Add convoy logic | | | OPEN |
| Update Game on submit | | | OPEN |
| Send Game Invites | | | OPEN |
| Get Move History | | | OPEN |
| Save Move History | | | OPEN |
| Add Timer | | | OPEN |
| Update Game on game end | | | OPEN |
| Update Game on withdraw | | | OPEN |
| Additional turn subtasks will be | | | OPEN |
| ⊟ User Interface tasks | | | OPEN |
| Logout Functionality | | Endg... | OPEN |
| Withdraw Functionality | | Endg... | OPEN |
| Add Map Functionality | | | OPEN |
| Add Player Pieces | | | OPEN |
| Additional UI elements will be b | | | OPEN |
| ⊟ Testing Tasks | | | OPEN |
| Register | | | OPEN |
| Login | | | OPEN |
| Start Game | | | OPEN |
| Resume Game | | | OPEN |
| Leave Session | | Endg... | OPEN |
| Logout | | Endg... | OPEN |
| Accept Game | | | OPEN |
| Chat: Private | | | OPEN |
| Chat: Public | | | OPEN |
| Send Draw Offer | | Endg... | OPEN |
| Respond To Draw | | Endg... | OPEN |
| Core game logic | | | OPEN |
| Place Order | | | OPEN |
| View Game Rules | | | OPEN |
| View Game Statistics | | | OPEN |
| View Move History | | | OPEN |
| Submit Orders | | | OPEN |
| Retreat Or Disband Order | | | OPEN |
| Build New Unit | | | OPEN |

### *4.1.1 Gantt Chart*



### *4.2 Schedule and Milestones*

| Milestones | Description | Milestone Criteria | Planned Date |
|---|---|---|---|
| M<br>0 | Start Project | Receive Requirements | <2019-02-05> |
| | The project requirements were given by the professor. | PRS or SRS reviewed<br>Stakeholders identified<br>Impl. Proposal reviewed | <2019-02-05> |
| M<br>1 | Start Planning | Receive Documentation Requirements | <2019-02-05> |
| | Create Vision Document, Project Plan, Use Cases, Test Plan, Program Flow Chart, and User Manuals | Complete Drafts of Documentation | <2019-03-07> |
| M<br>2 | Start Execution | Complete Documentation | <2019-03-07> |
| | Create a skeleton of the codebase based off the documentation | Code base framework fits the requirements of the project and is consistent with the documentation | <2019-03-08> |
| M<br>3 | Confirm Execution | Compile a working skeleton of the project | <2019-03-08> |
| | Create Prototype | Prototype has basic functionality | <2019-03-21> |
| M<br>4 | Start Introduction | Present prototype and get feedback from professor | <2019-03-21> |
| | Implement main functionality like the rules of the game | Coding of new functionality finished,<br>Revise documentation | <2019-04-00> |
| M<br>5 | Start Testing | | <2019-03-26> |

| | | | |
|---|---|---|---|
| | Complete testing key functionality such as the rules of the game | 85% test cases of key functionality passed | <2019-04-14> |
| M 6 | Release Product | | <2019-05-03> |
| | Continue testing the project | Diplomacy plays sufficiently and documentation reviewed | <2019-05-05> |
| M 7 | Close Project | Turn in Project | <2019-05-07> |

### *4.3 Development Process*

The development process we use for this project is agile. Agile is used because it fits the nature of this project where obstacles in implementing functionality of this software could come up. We are also creating many use cases and less strict documentation compared to models such as waterfall. Processes like spiral do not work because we do not have much time and we are not submitting multiple iterations of a project. Though agile is iterative, it focuses on building the main functionality of the application first then adding on additional functionality.

In addition, the project is also getting constant feedback from the professor and from the developers themselves. The project also uses plenty of user stories and use cases which is a key component of agile. It makes sense to adapt the project to the use cases and the feedback by the professor and the group.

The project will create a vision document, project plan, use cases, test plan, user manual, and a program flow chart then create a prototype. The prototype will be completed roughly two weeks after the documentation is due. The prototype will allow for good feedback from the professor on the progress of the project. The feedback will be used to improve the project.

### *4.4 Development Environment*

| Item | Applied for | Availability by |
|---|---|---|
| **Methods** | | |
| Use Case | Requirements capturing | M1 |
| Vision Document | Requirements capturing | M1 |
| Project Plan | Requirements capturing | M1 |
| Test Plan | Requirements capturing | M1 |
| Program Flow Chart | Requirements capturing | M1 |
| User Manual | Requirements capturing | M1 |
| **Tools** | | |
| Balsamiq Cloud | Design | M2 |
| Visual Studio Code | Coding | M2 |
| Electron Framework | Interoperability | M2 |
| Electron-toolkit | Packaging | M2 |
| **Languages** | | |
| UML | Design | M2 |
| HTML/CSS | Web interface for electron | M2 |
| Javascript | Building backend and frontend design and functionality | M2 |

## 5. Risk Management

A potential risk is getting a bad grade on the assignment for delivering a project that did not fulfill any of the requirements given by the professor. Another risk is if the product is not acceptable then no further or less funding may happen for future products. Another risk is losing the trust of further users that may have had faith in this project and was severely disappointed by the results. Less users would use the future projects and that would mean less revenue for the product.

The project may not be fully complete in the functionality of the game, but it will deliver full documentation and visions for finishing the game. If time permits the documentation will also layout some extra features that will improve that game in the future. Trello will be used to keep track of the WBS and deadlines.

## 6. Communication and Reporting

| Type of Communication | Method / Tool | Frequency / Schedule | Information | Participants / Responsibilities |
|---|---|---|---|---|
| **Internal Communication:** | | | | |
| Project Meetings | In-Person | Bi-Weekly and on event | Discussing potential problems. Also discussing project status and requirements. | Everyone on the team |
| Sharing of project data | Github, Google Drive | When available | All project documentation and code | Everyone on the team |
| Milestone Meetings | Slack / in-person meeting | Before milestones | Project status (progress) | Everyone on the team |
| Final Project Meeting | In-Person | M5 | Wrap-up and finalize project and documentation | Everyone on the team |

## 7. Delivery Plan

### *7.1 Deliverables and Receivers*

| Ident. | Deliverable | Planned Date | Receiver |
|---|---|---|---|
| D1 | Documentation | <2019-03-07> | Professor |
| D2 | Prototype | <2019-03-21> | Professor |
| D3 | Final Release of the Project | <2019-05-07> | Professor |

## 8. Quality Assurance

The Test Plan is documented in [3].

## 9. Configuration and Change Management

Any changes to the main functionality to the application will undergo a revision of documentation. The project will not stray too much from the requirements detailed by the professor. It does not seem there will be sudden changes to the requirements, at least for this kind of project. Though if there are major changes to the requirements then there would be major shifts in milestones and certain functionalities need to be relooked at before coding. Since we are doing agile, any changes to functionality can be accommodated by the documentation.

## 10. Security Aspects

The Diplomacy application won’t really have any personal information other than login info and a password. It is only for the sake of giving an alias to a person while they play with their friends or other random people. There is no matchmaking system so there is no chance of malicious people trying to obtain information from other people they are connected to. A lot of the security will be handled on the backend services such as Firebase keeping other people out.

## 11. References

[1] <Doc. No.> Vision Document

[2] <Doc. No.> Use Cases

[3] <Doc. No.> Test Plan

[4] <Doc. No.> Program Flow Chart

[5] <Doc. No.> User Manual

[6] <Doc.No.> Trello https://trello.com/b/BUREtiIU/cecs475-teamb

## 12. Revision

| **Rev. ind.** | **Page (P) Chapt. (C)** | **Description** | **Date Dept./Init.** |
|---|---|---|---|
| *1.0* | --- | original version | 02/13/2019 |
| 2.0 | --- | updated version | 05/05/2019 |
| | | | |