**Timelines - Work Division**

**Final Project Idea/Proposal**: Cooking app (recipe sharing platform and a network for those passionate about cooking).

**Team Model**: Democratic Team Approach. We will be implementing the *egoless* programming model where everyone on our team will have equal rights in terms of contributing to the development. All members of the group are considered equal in skill due to all of us being at the same stage of schooling (i.e. all of us being in this class). We realize that democratic team approach works the best when there is some sort of code review so other team members will review the code of others to find faults and bugs. This will help us identify the minor bugs (or even major bugs) that are difficult to find or slip by when tested by the developer who wrote that module. Having other members code review is highly beneficial because they bring a fresh and new perspective when testing a module that they had no part in writing.

While we do follow a democratic approach, we have a phase leader for each phase (requirements, analysis, design, implementation) that is responsible for conducting and organizing the meetings to discuss ideas and divide up components of that phase. We have a different phase leader for each phase (4 phases, 4 group members, 1 per each phase) so we can go into the specifics of what is needed for each phase without having a single-phase leader who is responsible for leading each phase meeting as that puts a lot of pressure on them.

**Project Timeline:**

Tuesday, Oct. 8th : Get acquainted with group members and discuss possible app ideas.

Thursday, Oct. 16th: Finalize application idea, allocate phase leader for each phase.

Tuesday, Oct. 21st: Meeting to discuss app specifics and necessities for the requirement phase. Work divided for this phase. Conducted by the requirement phase leader.

Wednesday, October 29th: Collect, assemble, merge, and finalize requirements phase document by phase leader.

Thursday, Oct. 30st: Meeting to find necessities for the analysis phase and division of work for this phase. Conducted by the analysis phase leader.

Wednesday, Nov 5th: Collect, assemble, merge, and finalize analysis phase document by phase leader.

Thursday, Nov. 6th: Meeting to find necessities for the design phase and division of work for this phase. Conducted by the design phase leader.

Wednesday, Nov 12th: Collect, assemble, merge, and finalize design phase document by phase leader.

Thursday, Nov. 12th: Meeting to address and go over any issues/bugs that are remaining for the implementation (development) process of the app.

Thursday, Nov 20th: Meeting to discuss the final report by the last phase leader.

Wednesday, Nov 26th: Collect, assemble, merge, and finalize final report document by the last phase leader.

Thursday, Nov 27th: General meeting held before app submission deadline to walk through entire app to ensure full functionality.

Sunday, Nov 30th: Practice app presentation.

**Work Division**: All work will be divided evenly among team members by application features rather than their individual skills. The reason being is that splitting up work by individual skills will end up creating a great misbalance in the amount of work done by group members and will end up putting a heavy amount of pressure on those who are more skilled thus causing a negative working environment. Splitting up by feature is highly beneficial since all of us are at an equal stage in schooling and we assume we all have the same skills. We understand that realistically in no team is work always done 100% equally, with this in mind we chose to split by feature and have team members lend a helping hand to those who are allocated to different features that are stuck or are having difficulty. This ensures that all group members will do an equal amount of work to the highest extent (of course it’s never 100% equal).

**Work Breakdown by Member**

**Ranadeep Polavarapu**

* Application backend
  + Core backend for recipe storage, rating, and user login features.
  + Create database and do the data modeling (i.e. creation of a schema).
  + Use a web framework and create a REST API that the app can query to perform all of its functions (recipe functions, login functions, sorting functions).
* Front-end login services (with Lee)
  + User Registration and User Login (JavaScript linkage).

**Lee Glendenning**

* Front-end Recipe services
  + Add and delete recipe.
  + Link this service to the API via JavaScript.
* Front end login services (with Ranadeep)
  + Create UI for login and registration.

**Ryan Burke**

* Front-end recipe sorting service
  + Sort by Recipe difficulty, Sort by cuisine, sort by prep time, sort up creation date.
  + Create UI for the sorting service.
  + Link this service to the API through JavaScript.
* Front-end recipe rating service (with Bruno).
  + Create UI for the rating a recipe service.

**Bruno Salapic**

* Front-end recipe like & dislike service.
  + Like a recipe or dislike a recipe.
  + Create UI for this service.
  + Link the like and dislike service to the API through JavaScript.
* Front-end recipe rating service (with Ryan).
  + Rate this recipe some “*x*” stars out of five (5).
  + Link this service to the API through JavaScript.