**CS673F16 Software Engineering**

**Team 3 Meeting Minutes**

**Date and Time:** 4/23/2018 9:30 pm

**Place**: Google Hangouts

**Participants:** Duncan, Ed, Ben, Zach, Yiannis

**Minutes taker:** Zach

**Timekeeper:** Yiannis

**Purpose:** Presentation planning

**Agenda**:

* Deployment and code: anything left?
* Presentation practice → run through whole thing

Order of presentation:

\* Quick high level overview/intro [explain concept of site 1 more time] (Duncan)

\* Team process and everyone's roles [quick] (Zach)

\* Requirements (Yiannis)

\* Design/Architecture [Stuff like class diagrams, schema, algorithms] (Zach & Ed)

\* Implementation [Rails architecture and AWS] (Duncan & Ben)

\* Testing/Quality metrics (Ed)

\* Security (Ben)

\* Product demo (Ed & Yiannis)

\* Reflection on team process and areas for improvement (Zach)

\* Conclusion (?)

**Notes:**

* Ed to work on final merging/deployment before thursday
* Yiannis/Ed to test functional requirements
* Duncan to make Readme next week
* Zach to put documentation on github
* Everyone sign the documents
* Pivotal tracker → everyone check out

**Date and Time:** 4/16/2018 8:00 pm

**Place**: Google Hangouts

**Participants:** Duncan, Ed, Ben, Zach, Yiannis

**Minutes taker:** Duncan

**Timekeeper:** Zach

**Purpose:** Iteration 3 wrap up/Presentation planning

**Agenda**:

* Review everyone’s status
* Project deployment
* Documentation
  + SPPP
  + STD
  + SDD
  + Readme.md
  + Pivotal tracker final version
* Final presentation brainstorm
  + Topics
    - Final software demo (**Ed & Yiannis)**
      * Create user
      * Create lists, items
      * Create group
      * View other users list
      * Mark things as purchased
      * BOOM MAIN REQUIREMENTS ALL HIT
    - Requirement analysis (**Yiannis**)
      * Final iteration stories implemented
      * All stories implemented
      * What we laid out as nice to haves, must haves etc
    - Final software architecture (Zach & Duncan)
      * Design patterns
      * Class diagrams
      * Database schema
    - Environment Used (Ben - AWS, Duncan - Rails)
      * Ruby, rails, difficulties with
      * AWS deployment
    - Security (Ben)
    - Testing (Ed)
      * Unit testing
      * System testing?
      * Testing coverage
      * Quality control and metrics (from STD)
      * Defect management
    - Process improvement (Zach)
      * What our process was
      * What worked, what didn’t
      * What could’ve been improved
      * Refactoring
    - Conclusion (Duncan)
      * “Lessons learned”?

**Discussion:**

Statuses

* Zach → working on css, should have pull request in soon
* Ed → finished user addition to groups, in Yiannis’s branch
* Yiannis → pull request in for flash messages/display, needs review
* Ben → Still writing tests, production environment and deployment script mostly written. Getting error, Ed to help him get Tiny TDS working
* Duncan → last few CSS fixes need to get finished

**Date and Time:** 4/9/2018 9:30 - 10:15 pm

**Place**: Google Hangouts

**Participants:** Duncan, Ed, Ben, Zach, Yiannis

**Minutes taker:** Ed

**Timekeeper:** Duncan

**Purpose:** Iteration 3

**Agenda**:

* Authentication
* Deployment
* Groups
* CSS
* HTML Flash
* Documentation

**Discussion:**

* Ben’s updates
  + Mainly focused on testing
  + Looking into security testing gems/libraries
  + Working on setting up production environment
  + On track to finish by deadline
* Zach’s updates
  + Looking to focus on test coverage, mimic behavior of list page in groups page
* Ed’s updates
  + User suggestion wired into groups page
  + Want to transition from using User IDs and move to names
* Duncan’s updates
  + CSS updates to Lists page
    - Demo
* Other items
  + Readme for GitHub

**Date and Time:** 4/2/2018 9:30 - 10:15 pm

**Place**: Google Hangouts

**Participants:** Duncan, Ed, Ben, Zach, Yiannis

**Minutes taker:** Ed

**Timekeeper:** Yiannis

**Purpose:** Iteration 3 Planning

**Agenda:**

* Proposed Iteration 3 Breakdown of work
  + Ben
    - Authentication finalization
    - Deployment to AWS
    - Security testing?
  + Ed
    - User adding to Groups/search
    - SQL AWS deployment (work with Ben)
  + Duncan
    - Website CSS Overhaul (all pages)
    - Ranking list elements?
    - Page footer
  + Yiannis
    - HMTL Flash divs and feedback. (Done)
    - Input validation and feedback for text boxes (Done)
    - List element Purchased status visibility for owner/viewers (Done)
    - All actions on one page for creation/edit/delete? (not going to do. Dont think this is necessary. Looks cleaner the way it is now. Maine page, i think, would look too cluttered when trying to add item with this implementation).
  + Zach
    - Group page CRUD finalization
    - Group page unit test
    - Group page CSS
  + Others to possibly assign:
    - Gift suggestions
    - Do we want to scrap amazon integration? I believe it costs money
* Scheduling
  + Do we want to say all coding complete one week prior to presentation, and then spend last week for presentation? Final presentation is 30-45 minutes and more involved

**Discussion**:

* Proposed Iteration 3 Breakdown of work
  + Ben
    - Authentication finalization
      * Email verification
        + Concerns around how to work on this in a dev-only environment
    - Deployment to AWS
      * Concern with connections to SQL in this environment
    - Security testing
      * General security testing
      * Unit testing
  + Ed
    - User adding to Groups/search
      * Need to wire this into the Groups page
      * Need to decide how to handle users that are members or few or no groups
    - SQL AWS deployment (work with Ben)
      * Config already partially in database.yml
  + Duncan
    - Website CSS Overhaul (all pages)
      * Consistency
    - Ranking list elements
      * Next priority after CSS
    - Others, if time
      * All actions on one page for creation/edit/delete?
      * Page footer
  + Yiannis
    - HMTL Flash divs and feedback.
      * We’re passing the data, but need a place to display it
      * Use login page as an example
    - Input validation and feedback for text boxes
      * Ideally want to handle specific text fields
      * Begin/Rescue if all else fails
    - List element visibility for owner/viewers
      * Hide management items for viewers
    - Classifying users
      * Not a high enough priority
  + Zach
    - Group page CRUD finalization
      * Remove users from group
    - Group page unit test
      * For view/controller
    - Group page CSS
      * Work with Duncan to harmonize with rest of site
  + Others to possibly assign:
    - Gift suggestions
      * Have user recommend item to owner?
    - Tying lists to events
      * Low value add
    - Do we want to scrap Amazon integration?
      * Associated cost?
* Scheduling
  + Do we want to say all coding complete one week prior to presentation, and then spend last week for presentation? Final presentation is 30-45 minutes and more involved
    - Presentation is on Thursday, April 26
      * 30-45 minutes, covers entire project
      * Everyone must present
    - All code checked in by EoD Thursday, April 19
      * Gives 1 week to make presentation, finish documentation, etc.

**Date and Time:** 3/27/2018 9:30 - 10:15 pm

**Place**: Google Hangouts

**Participants:** Duncan, Ed, Ben, Zach, Yiannis

**Minutes taker:** Ben

**Timekeeper:** Zach

**Purpose:** Presentation Prep

**Agenda:**

* Quick review of status (5 minutes)
* Presentation
  + Presentation branch? Or groups branch with latest changed rebased in from list\_refactor branch?
  + Zach and Ben to prepare - what to include
    - User stories implemented
    - Demo
      * Users demo
      * Authentication
      * Groups
      * CSS redo
      * RSpec replacement of testing
    - Schema diagram, UML diagram → Ed how were these generated last time?
    - Testing results
    - Iteration 3 preview - besides our “nice to haves” (suggestions/site integration) I think we look pretty good
      * What do we want to say here?
      * Create types of lists
      * Actual CSS redo of Users/Groups pages
      * Rank/Organize list items
      * Reaches:
        + Item suggestions
        + Vendor site integration
* Documentation
  + SPPP
  + SDD
  + STD
  + Risk Management
  + Pivotal Tracker
  + Previous iteration all in github

**Discussion:**

End of iteration 2:

* Create iteration 2 branch on Github
* Yiannis - update requirements in SPPP and in Pivotal
* Zach and Ben - finish changes and submit pull requests by evening of 3/28
* Ben - Update SPPP Config Management and add key auth features/changes as chores on Pivotal
* Ed - generate testing data and update tests report document
* Presentation
  + talk about user stories implemented, talk about switching to RSpec, get updated figures (UML, testing results, etc)

Iteration 3:

* Focus on improving the appearance of the app
* Deploy to production environment
* Vendor integration as a reach

**Date and Time:** 3/19/2018 9:45 - 10:30 pm

**Place**: Google Hangouts

**Participants:** Duncan, Ed, Ben, Zach, Yiannis

**Minutes taker:** Zach

**Timekeeper:** Ben

**Purpose:** Iteration 2 progress

**Agenda:**

* Users/Lists
* Authentication/Deployment
* Groups page
* Pivotal tracker
* Presentation/deliverables for next week

- Documentation we need:

* + - SPPP v3
    - SDD document (class diagram & design documentation)
    - STD document (Testing documentation)
    - Risk Management documentation
    - Requirements all in Pivotal Tracker and updated to current status
    - Presentation for Thursday

**Discussion:**

* Users/Lists (Yiannis and Duncan)
  + Need to make lists deletable.
  + Modify needs to work for lists and list items
    - Get rid of modify list buttons
  + Separate list headers from list items
    - CSS adjustments
  + Collapse list items once a list is opened
* Authentication/Deployment/Sessions (Ben)
  + Will try to have pull request in by thursday
    - Persistent sessions
    - Change how the hashing works
    - Password reset
* Groups page (Zach and Ed)
  + Need to get creating and adding user functionality to a group
  + Once adding and creating is done, make sure to add deletion
  + Ed has added a user suggestion querying function
* PivotalTracker
  + Discussed all of the tasks in the ice box
* Documentation
  + Hold til next week to see where we are at
  + Zach and Ben to present
* Stay in contact in slack

**Date and Time:** 3/12/2018 9:45 - 10:15 pm

**Place**: Google Hangouts

**Participants:** Duncan, Ed, Ben, Zach

**Minutes taker:** Zach

**Timekeeper:** Ben

**Purpose:** Iteration 2 progress

**Agenda:**

* RSpec
* Users/Lists
* Authentication/Deployment
* Homepage/CSS
* Groups page
  + How to delete groups
  + Consent to join group? Or just join?

**Discussion**

* RSpec
  + Functionality is extremely similar but more comprehensive than minitest
  + More matching criteria but asserts still work
* Users/Lists
  + Yiannis working on, to provide update Thursday
* Authentication/Deployment
  + Ben to continue looking into persistent sessions
* Groups Page
  + Do we need users to consent to join group?
    - For first iteration just be able to search and add people
  + We need users to have an identifier in order to search and display for the groups page. Duncan will look into this week and add.

**Date and Time:** 3/5/2018 8:00 - 9:00 pm

**Place**: Google Hangouts

**Participants:** Duncan, Yiannis, Ed, Ben

**Minutes taker:** Duncan

**Timekeeper:** Zach

**Purpose:** Iteration 1 reflection. Iteration 2 planning

**Agenda:**

* Iteration 1 reflection
  + Development Process
  + Requirements → how to better going forward
  + Communication
  + Better TDD
* Iteration 2 Planning
  + Authentication pt 2
    - Password complexity Validation
    - Forget password email
    - Login page
    - Persistent sessions, timeout
    - Examine session setup further, check for security vulnerabilities
  + Users page
    - Use user\_id from login to connect to and show what lists correspond in database
    - Params method to instantiate
    - CSS
      * Replace Modify/Delete words with pencil, X buttons
      * Style "Create new list" button
      * Limit div width for wishlist items, surround with grey (remove app width restriction)
  + Lists
    - Refactor List#Create and List#Save
    - Add a List\_Item controller and move relevant actions into that
    - Params method to instantiate
  + CSS overhaul/scaling
    - Homepage
      * media query - css for fonts, image
      * 3 information divs
      * Link start button to login
      * Add unit tests for all links
      * Eliminate width restriction
    - Auth/signup pages
      * Scale form for mobile
      * Incorporate style of website (grey/white/blue/white)
    - Users/Lists
      * Incorporate Style of homepage
      * Limit list item height?
    - Footer
      * Do we want a footer with copyright, contact info?
    - Flash
      * Divs to display "Logged in!" etc
  + Groups
    - Group page (My Groups --> similar to my lists)
      * Create Group
        + Find other users --> Email or search function?
      * Select lists to share with group
      * View Lists (with purchased button unseen by original person). Same View as List, extra bool field
        + Purchased == updatable
  + RSpec/Testing
  + User Classification
  + Rank/Organize wishlist Items

**Discussion:**

* Iteration 1 reflection
  + Previous iteration had a lot of work slowed as it depended on others work. Mostly this was a result of it being the main app architecture versus a process issue. No change for now
  + Requirements → use pivotal tracker more directly versus discussion.
    - If the work is for a story completed in iteration 1 mark as a chore, if for an iteration two story, add as a task
  + TDD
    - Ensure existing unit tests pass before you check stuff in
    - Ensure what you develop has test coverage, whether red/green methodology or after the fact
* Iteration 2 Planning
  + Authentication pt 2 → Ben
  + Setting up deployment/production environment → Ben
  + Homepage CSS → Duncan
  + Users page → Yiannas
  + Lists → Yiannas
  + CSS overhaul/scaling → Duncan
  + Groups → Zach/Ed
    - Update wireframe to get rid of “List” and say the name of the group → Duncan
    - Two parts: (Zach and Ed to each take one)
      * The view itself, linked to the users page
      * Searching/adding group members
  + RSpec/Testing
  + User Classification - nice to have for this iteration
  + Rank/Organize wishlist Items - nice to have for this iteration

**Date and Time:** 2/26/2018 9:30 - 10:30 pm

**Place**: Google Hangouts

**Participants:** Duncan, Yiannis, Ed, Ben, Zach

**Minutes taker:** Ed

**Timekeeper:** Yiannis

**Purpose:** Iteration 1 wrap-up. Finishing touches, presentation prep, documentation

**Agenda:**

* Finish login/auth and integrate with Yiannis's branch
* Merge Yiannis's branch and analyze what else needed for initial list functionality
* Finish CSS cleanup homepage
* Reorg github with top level documentation folders
* Good, not mandatory:
  + CSS v1 on users/login/signup page
  + test coverage added for users page, login page, homepage
* Documentation we need:
  + Presentation for thursday (Ed and Yiannis to present)
  + SPPP v2
  + SDD document (class diagram & design documentation)
  + STD document (Testing documentation)
  + Risk Management documentation
  + Requirements all in Pivotal Tracker and updated to current status

**Discussion:**

* Login/Authentication, User CRUD (Ben)
  + Login is working
  + Still adding unit tests, code cleanup to do
  + Helper class for controllers to add authentication
  + Add logged\_in? To ApplicationController to execute with each command
  + Pull request tomorrow for review
* List and List Item CRUD (Yiannis)
  + Add, delete, modify contents of List
  + Add, delete list items
    - No modify yet
  + Rename List item description/comments to name/description
  + If site link is image, then image will display for list item
    - If site link is site link, then no image for now
  + Additional branch made by Ed to move from custom CRUD functions to ActiveRecord
  + Duncan to assist with streamlining routing, etc.
  + Refactoring brainstorming
    - Try and do as much as we can on one page, rather than moving between pages for each action
      * Reload page on data commits, etc.
* Demo for Iteration 1 presentation
  + Create separate demo branch to get a functional version of Ben/Yiannis’ code before merging to master
    - Code cleanup, refactoring, can be done prior to merge to master
  + Yiannis to create branch
  + Will merge into master following refactoring in Iteration 2
* Home page demo (Duncan)
  + Minimalist style with blue/red accents
  + Shouldn’t have conflicts (mostly CSS updates)
    - Potential conflicts with Ben’s branch, but should be easy to resolve
  + Envisioning 4 main views with options underneath them
    - Home view
    - Login view
    - Groups view
    - List view
  + Pull request available Tuesday (2/27) night?
* To Dos
  + For Thursday (3/1)
    - Reorganize github with top level documentation folders
    - Documentation we need:
      * SPPP v2
        + Zach to update
      * SDD document (class diagram & design documentation)
        + Ed to update
        + Duncan, Yiannis, Ben to add bullet points, etc.
      * STD document (Testing documentation)
        + Ed to update
        + Needs branch to generate test cases, metrics, etc.
      * Risk Management documentation
        + Have risks outlined in meeting minutes

Need to be added to risk management document

* + - * + Zach to update
      * Requirements all in Pivotal Tracker and updated to current status
        + Need to update requirements
        + Bugs to get from testing
        + Chores for minor tasks (code cleanup, etc.)
        + Everyone to go through and check off tasks completed
      * Presentation for Thursday (Ed and Yiannis to present)
        + 10-15 minute presentation
        + Focus on site demo, code
        + Touching points

What did we complete? (Demo)

Difficulties, how did we resolve them

Plans for Iteration 2

* + For Iteration 2
    - CSS v1 on users/login/signup page
    - Routing refactoring
    - test coverage added for users page, login page, homepage
    - Group functionality
    - Marking gifts as purchased
    - To discuss more Thursday

**Date and Time:** 2/19/2018 8:30 - 9:30 pm

**Place**: Google Hangouts

**Participants:** Duncan, Yiannis, Ed

**Minutes taker:** Yiannis

**Timekeeper:** Duncan

**Purpose:** Iteration 1: Models, Authentication, Homepage, Users

**Agenda**:

* Application overview, environment and debugging (Duncan, All)
* Models and Database setup (Ed)
* Authentication, signup page (Ben)
* Requirements
* Next steps:
  + User Profile on login
  + User page (user#show) → CRUD operations
    - Best way to divide work
    - Need Lists/List Items model to load.
  + Homepage
  + Documentation needed for this iteration:
    - SPPP
    - SDD
    - STD
    - Risk Analysis

Discussions:

**Code Overview**

**General Code Layout:**

* model-view-controller in the app folder.
* All controllers inherit from application\_controller.
* Can also create helpers for corresponding controllers.
* Controller has corresponding js file and css file.
* Using ajax calls to interact with database in jquery.
* Config dropdown -> routes folder. counterpart to controller. Essentially simplifies urls
* Tests folder is for unit-tests.

**DEBUGGING:**

* For debugging one could possibly integrate ruby debugger into Chrome.
* Use regular Chrome dev tools for web page debugging.
* Ruby debugging -> “rails start webrick” will start debugger.
* The word "debugger" in code acts as a breakpoint.
  + At a breakpoint one can inspect variables and run methods as well in the terminal.

**DATABASE:**

* Models, when generated using the rails generate model command, will generate several different files.
  + Definition of model
  + migration script (creates database file),
  + fixtures file (for pre populating data for unit tests)
  + unit test file.
* Useful Commands:
  + rake db:drop (drops database)
  + rake db:create (creates them new)
  + rake db:migrate (will bring you up to date with migration scripts)
* Environment variables for database credentials in code correspond to windows env variables. Set them there. Search for environmental variable in windows search.
  + These are used in database.yml to provide username, password, etc. and not providing them to the public
  + After creating new environment variables, restart your IDE to make use of them
  + How-to document to be provided to team and uploaded to Github
* Database connection: make sure you have login defined so you can connect to database.
  + Give it the sysadmin server role so you can connect to everything.
* Currently email and password are in memory and stored unhashed.

**General Notes**

* Ed send screenshots of environment variables out along with notes on database.
* Look into CI?
* Ben send out notes on auth.
* Duncan continue to fix homepage.
* Yiannis user's controller page. Also fix second non functional req.
* Yiannis work on show and new method well as list controller.
  + Note: Create user in local database, create some lists and list items.
* Duncan send Ruby Tutorial.
* Ed merge in models when done with testing.

**Date and Time:** 2/12/2018 9:30 - 10:30 pm

**Place**: Google Hangouts

**Participants:** Duncan, Ed, Ben, Zach

**Minutes taker:** Ben

**Timekeeper:** Ed

**Purpose:** Iteration 1 Requirements, models, next steps

**Agenda:**

* Review pages wireframe and design proposals (Duncan)
* Review user schema and models (Ed)
* Review Google authentication (Ben)
* Discuss HW2: We each need to make stories to add to PivotalTracker. Let's divvy up our requirements so we don't overlap
* Next steps (this week):
  + Homepage
  + User model
  + User View
  + SQL Express setup → Everyone complete?
* Iteration 1 Deliverables & Discussion Points:
  + Requirement Analysis (PivotalTracker)
  + Implementation:
  + Coding Standards, Code reviews, Testing, Working software
  + SDD, STD, SPPP Documentation
* How do we want to divide coding/documentation/review etc? We probably aren't going to have 6 people developing this simultaneously at this point. Those less experienced in rails/web focus on documentation/code review at this point, more coding in later iterations?
  + Needs:
    - Documentation
    - User, Homepage design/implementation
    - Code Reviews
    - Other?

Discussions:

* Review wireframes for UI design
  + See Duncan’s links in Slack channel
* Review DB schema
  + See Ed’s SQLDBM.com link in Slack channel
  + How will using Google auth change our DB schema?
    - Don’t need a password field, just need some primary key for the user table which can be obtained from the auth token (email address would work well - Ed mentioned we should maybe hash it to avoid storing plain text email addresses)
* Discuss Google Auth
  + Figure out how to get a domain name for the application to be able to get an API key.
  + Make sure it will be possible to run Google auth when developing locally
* Discuss Homework 2
  + Duncan will assign stories to avoid conflicts
* Next week
  + Ed will finalize DB models, Zach and Duncan will work on user view, Ben will work on authentication, Yiannis will focus on docs.
* Misc notes
  + Create feature branches in Github. Create a pull request and mention it in Slack channel to request code review.
  + For iteration 1, testing and design documents must be updated.

**Date and Time:** 2/5/2018 9:30 - 10:30 pm

**Place**: Google Hangouts

**Participants:** All members

**Minutes taker:** Zach

**Timekeeper:** Yianny

**Purpose:** SPPP Review, Presentation Review, Project first steps

**Agenda:**

1. SPPP Walkthrough - review each section
2. Presentation preparation -- what to include
   1. Introduction - Our motivation and vision of the app
   2. Overview of the current marketplace
   3. Requirements, and how we plan to address them
   4. Our technology stack
   5. Our process
      1. Development
      2. QA
      3. Other?
   6. Homepage wireframe, DB schema slides
   7. Anything else?
3. Project next steps - Assign action items
   1. Routing and homepage setup
      1. How do we expect the homepage to look? Theme suggestions?
   2. User data model
   3. Database connectivity and schema proposal
   4. Login and authentication
4. Started walkthrough of the SPPP
   1. Overview and related products section looked good and was approved.
   2. Talked about the differences between functional and nonfunctional requirements.
   3. Discussed the management plan, need to make sure to constantly update the risk management plan as well as the schedule to add demos and feature deadlines.
   4. The QA plan was discussed and was decided that it was good for all members to read through the entire plan to make sure everything looked good.
   5. Configuration Management plan was discussed, will use AWS with an Elastic Beanstalk environment. Be able to run everything locally for development reasons.
   6. Group decided to go over their own SPPP sections to make sure there are no typos and update revision history for Thursday’s deadline
5. Presentation talk
   1. Duncan and Zach will present
   2. ~8 slides
      1. Intro to app - duncan
      2. Marketplace, where we stand - duncan
      3. Main requirements and which iteration they belong to - zach
      4. Ruby and app setup - duncan
      5. Our process - zach
      6. Preview of app - duncan
         1. Wireframe for homepage
         2. Ruby enforces the MVC architecture
         3. Keep it high level
6. Next steps:
   1. Duncan and Zach will work on slides for Thursday presentation
   2. Ed and Ben will work on DB integration and setup
   3. Yiannis will update requirements and add to pivotaltracker and tie in github to pivotaltracker

**Date and Time:** 1/28/2018 4:00 - 5:00 pm

**Place**: Google Hangouts

**Participants:** All members

**Minutes taker:** Duncan Morrissey

**Timekeeper:** Zach Lister 4:06-4:43

**Purpose:** Initial kickoff/Requirement meeting

**Agenda:**

1. Team Roles - Two roles open (Yiannis?) - Requirement Leader and Implementation Leader.
2. Development Environment - Ruby/Rails easier to run on Unix systems, but is still very good on windows today
3. Coding Standards - https://github.com/bbatsov/ruby-style-guide + RuboCop in IDE.
   1. Do we want a formal pull request/approval system for merging in changes? Or if you're merging something major then request others reviews at your discretion?
4. Communication
   1. Slack
   2. Pivotal Tracker - Has anyone used this before? Requirement leader to learn/teach team
   3. Formalized processes beyond weekly meeting? For example, as professor suggested - daily updates what have you worked on, what will you work on.
5. Product Name - Brainstorm
   1. Not available: Wishlistr, Giftster (as these already do what we're trying to)
   2. Tentative ideas: The Wish List, Group List, Friend Registry, Wish List Registry…
6. Major Functionalities and Requirements
   1. What are our requirements?
   2. What Iteration we expect each to fall into
   3. What can differentiate our app from the apps already online?
      1. [https://www.wishlistr.com](https://www.wishlistr.com/)
      2. <https://www.myregistry.com/>
      3. In house apps (Walmart, Amazon, Target)
7. What are possible risks of our project?
   1. Team member drops course or is unresponsive
   2. Team members have difficulty learning ruby/new development stack
   3. Lack of JavaScript experience on team creates poor UI or significant time sink to team
   4. Others?
8. Documentation
   1. SDD - Design Leader manages, others add/review?
   2. Testing Document - QA leader manages, others add/review?
   3. SPPP - Duncan manages, others add to their relevant sections
   4. Project Proposal, Management Plan QA Plan, Configuration Plan
9. Our Process
   1. Agile, weekly meetings
   2. How do we want to do testing/unit testing? QA leader have ideas?
10. Phase 1 deliverables:
    1. SPPP
       1. Requirements
       2. Related work and how we differ
       3. Process model - agile…
       4. Research hosting options
       5. Risk management
       6. Monitoring
       7. QA Plan
       8. Standards - Review Process, testing, defect management
       9. Configuration management

**Discussions:**

* Team Roles
  + Yiannis taking Requirement + Implementation leader roles
* Development Environment
  + Develop on Windows, VSCode
* Coding Standards
  + Use the base ruby style guide + RuboCop to ensure. As we begin to create models of our project iron out further details
* Communication
  + Slack for all questions/communication, meetings on Sundays at 4-5 pm + class time
* Product Name
  + The Wish List
* Major Functionalities and Requirements
  + View Requirements in terms of “As a \_\_\_\_, I want to \_\_\_\_”
  + From the Wishlist Creator’s perspective:
    - Create/Modify Wishlist Items
    - Share a wishlist
    - Rank wishlist items
    - Link to wishlist items (on Amazon, Walmart etc)
    - Determine who can access a wishlist
    - Receive suggestions from the app on other products someone may like
  + From a Wishlist viewer’s perspective
    - Search for a wishlist/items
    - See what items have already been purchased
    - Suggest wishlist items
* Risks
  + If a team member drops course, divvy out their responsibilities as our timecard would indicate others have availability to absorb
  + Difficulty with ruby - allocate more development heavy tasks to those experienced in the language earlier in the project.
  + Javascipt/UI - use bootstrap + jquery to follow industry wide straightforward standards for setup
  + Concern: Does SQL Server Express work with Rails well? - Ed to look into any issues here
* Documentation
  + SDD - managed by Environment/Implementation leaders
  + Testing doc - managed by Ed, others to add to as needed
  + SPPP - managed by Duncan - others add to their relevant sections
* Our Process
  + Weekly meetings
  + Pull requests + reviews for new code added to the project
  + Unit tests for all components, written tests for features

**Action Items:**

* Set up development environments - all
* Ruby set up/IDE setup information - Duncan to send out
* Yiannis to look into Pivotal Tracker
* Ed to look into SQL Express + Rails
* Duncan to create repository for app
* All - read on ruby and do ruby/rails training. Mike Hartl’s tutorial is very good.