Scrum Prototype

# Cloud Group Roles

|  |  |  |
| --- | --- | --- |
| **Name** | **Email** | **Role** |
| Michael Meding | [Michael\_Meding@student.uml.edu](mailto:Michael_Meding@student.uml.edu) | Software Developer |
| Ramon Meza | [Ramon\_Meza@student.uml.edu](mailto:Ramon_Meza@student.uml.edu) | Product Owner |
| Tylor Reeves | [Tylor\_Reeves@student.uml.edu](mailto:Tylor_Reeves@student.uml.edu) | Scrum Master |

# Artifacts

## First Meeting

Our initial requirements from our feasibility study were as follows,

1. Whatever database(s) we choose to use must be able to properly and securely store all pertinent user information, such as login information, application settings, and custom keybindings, if any.
2. Our database(s) must be able to keep sensitive user information such as, at the very least, the user’s password encrypted to ensure the confidentiality of said user information.
3. The database(s) must be able to quickly and efficiently communicate with the website, web UI, and potentially the phone companion app.

Our first meeting took place just after class on Thursday April 2nd. We met for approximately 20 minutes and discussed how we would be doing the basic structure of our cloud backend. We decided to use MySql for the underlying database and PHP as our interface layer. To communicate with the data aggregator group we will be using a public RESTful interface for testing before moving it to an encrypted interface for deployment. We also decided upon our database schema which to start is only 3 simple tables which is detailed in our backlog.

## Sprint Planning

We held our sprint planning meeting on Friday April 3rd via Skype. In this meeting we discussed how we would be splitting the work for this task. As time is limited and we are the smallest of all of the groups we thought it wise to set modest goals for each of us. This allowed us enough time to get our work done and also time to work on putting this documentation together.

### Michael Meding

* Design simple frontend for visualizing database
* Implement this using available technologies to save time

### Tylor Reeves

* Implement database schema using Mysql
* Connect MySql database with PHP
* Assist Ramon Meza with his implementation of the PHP backend

### Ramon Meza

* Design methods for handling database calls in PHP
* Implement these methods and allow for public interface access
* Coordinate with Tylor Reeves in this implementation

## Daily Meetings

### Meeting #1 - Tuesday March 31

We held a meeting in the classroom after class for about fifteen to twenty minutes to discuss some of the requirements in a little more detail with each other and to start picking roles for each of the three members of our group. We decided that who would do what, as stated at the top of this document. We planned our next meeting to be the coming Thursday after class, as that was the most convenient for all of us and would give us time to work on our portions of our project before we met again.

### Meeting #2 - Thursday April 2

Again, we had our meeting after class as it was the most accessible time for all of us and our differing schedules. At this meeting we went over what we had accomplished on the day we hadn’t met, that past Wednesday. We also started to try to conceptualize what we wanted most from our website/app and how we could abstract the workload out better. We decided that the best day to meet again would be Friday, April 3, that way we can have the weekend to work on the project and try to make progress on parts of the project we thought would be difficult to overcome.

### Meeting #3 - Friday April 3

We met using Skype instant messaging on Friday night. We talked about the layout of our website that we had worked on yesterday (Thursday) and brainstormed ideas on how to streamline the user interface. We decided, with the guidance of Professor Cao, that we should take a look at the layout and design choices made by MyFitnessPal, a successful health website/app. After dissecting the website design, we updated our design and tested it out ourselves. We decided we were happy with what we came up with and proceeded to make plans for our next meeting. We scheduled our next meeting to be after class on Tuesday April 7.

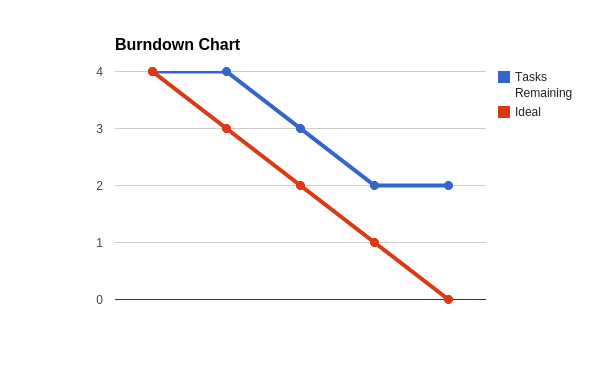
### Meeting #4 - Tuesday April 7

This meeting we spent our time discussing how we would implement our website application. We spent a majority of the time discussing which host for a site would offer us the best service for what we needed. We also decided that our main focus would be on databasing, and thus decided that PHP and MySQL were a must for our development. We made plans to meet the upcoming Thursday, and also decided that we should have a semi-functional webpage up and running before then. We also defined semi-functional as no styling, just being able to navigate through a blank website structure with only simple interactions.

### Meeting #5 - Thursday April

The final meeting was held over skype. Most of our discussion was focused on the website which had been set up in the last few days. We discussed the planned database schema and how the website would interact with the SQL database using PHP queries, as well as the overall design of the website. We made plans to design and implement additional interfaces for our website that users might interact with, such as account creation, a login page, and a page where users can set their application preferences. We also planned to have the SQL database setup complete by the end of the day.

# Burndown Chart



# Backlog

## Cloud Storage Backlog (this needs to be expanded)

1. Storage of user pertinent information such as username, password, email, and permissions.
2. Ability to store and retrieve the individual user settings for the keyboard such as keyboard color and theme and the keyboard arrangement such as QWERTY or DOVARK.
3. Having the ability for the user to store, update, and delete custom keybindings for shortcuts on the system. Perhaps even having the keybindings mapped as specific actions of the user as opposed to key presses.
4. This would need to have the ability to be easily accessed as the keyboard application gets updated. This way users may retain all information with new app updates.
5. Having the ability for users to reset their password using a no-reply email account created using the free hosting website’s cPanel.
6. Having passwords be stored in an encrypted format in the SQL database to protect user information.