**1. Project vision**

**1.1. Backgrounds**

Noah Bradley - Web Development / Mobile App Development

Major: CS

Languages: Java, C, C#, C++, Python, Ruby, HTML/CSS, XML, FXML, SQL

Jizhou Zhang - Web Development / Solution Tester

major: CS

Languages: Java, Javascript, C++, C#, Python, C, CSS, Html,

PHP

Andrew Dever - Database Design

Major: CS

Languages: Java, SQL, HTML/CSS, Visual Basic, Python, C#, C, C++, Javascript, Power Query M, PowerShell, TI-BASIC

Isaac Watson - Web Development / Database Implementation

Major: CS

Languages: Java, Javascript, C/C++, C#, HTML/CSS, SQL

Bryce Waters - Server Administration

Major: IT

Languages: Java, HTML, CSS, PHP, SQL, PowerShell, Bash

**1.2. Socio-economic Impact, Business Objectives, and Gap Analysis**

Socio-economic impact: Keeping track of the monetizing unit depends on

the number of people registering a Memcomb account, using

their account to post content, matching with other people, liking

Other people.

Business Objectives: Our objective is to have a 2.5% increase in profit

Per week.

Gap Analysis: Comparing the profit performance between the current

week and past week. Also compare the profit performance between the current month and the past month. Also compare the profit performance between the current year and last year. Also keep track of the number of users increased or decreased monthly as well as how many people go online each day.

**1.3. Security and ethical concerns**

Security concerns: Vulnerability access to user’s full phone numbers and

email addresses. Capability of locating a user’s location.

Vulnerability to virus attacks. Capability of hacking and accessing another user’s electronic and account

Ethical concerns: Cyberbullying concerns, Inappropriate content,

Friending someone who probably isn’t who user think they may   
 be.

**1.4. Glossary of Key Terms**

Bio

Blog

Chat

Comment

Direct Message

Friends

Followers

Location

Match

Posts

**2. Project Execution and Planning**

**2.1. Team Information**

Oftenly hold meetings during Mondays or Wednesdays for 3 hours

Whenever we do not have class. All are researchative and

Resourceful in searching for solutions to fix technical difficulties when tools were not functioning properly.

**2.2. Tools and Technology**

* GroupMe
* Discord
* Github
* Visual Studio
* Trello
* ASP.net
* MySQL

**2.3. Project Plan**

1. Discuss project idea and requirements, tools, and languages.
2. Begin working on database and interface components (higher level)
   1. Discuss design, colors, patterns, common shapes for interface
   2. Discuss database relationships, fields, then connect it to interface
3. Begin implementing interface components, login screen, account settings, and controllers to accompany them
4. Discuss the overall timeline, algorithm to make it work, how it relates to the database
5. Research “An interface for allowing the user to capture event data in form of audio, video, images and text data.”
6. Database design. What tables we will need (user account, event data, etc.)?
7. <https://help.github.com/en/desktop/getting-started-with-github-desktop/configuring-git-for-github-desktop>

**2.4. Best standards and Practices**

**3. System Requirement Analysis**

**3.1. Function Requirements**

**3.2. Non-functional Requirements**

**3.3. On-Screen Appearance of landing and other pages requirements.**

**3.4. Wireframe designs**

**4. Functional Requirements Specification**

**4.1. Stakeholders**

**4.2. Actors and Goals**

**4.3. User stories, scenarios and Use Cases**

**4.4. System Sequence / Activity Diagrams**

**5. User Interface Specifications**

**5.1. Preliminary Design**

**5.2. User Effort Estimation**

**6. Static Design**

**6.1. Class Model**

**6.2. System Operation Contracts**

**6.3. Mathematical Model**

**6.4. Entity Relation**

**7. Dynamic Design**

**7.1. Sequence Diagrams.**

**7.2. Interface Specification**

**7.3. State Diagrams**

**8. System Architecture and System Design**

**8.1. Subsystems / Component / Design Pattern Identification**

**8.2. Mapping Subsystems to Hardware (Deployment Diagram)**

**8.3. Persistent Data Storage**

**8.4. Network Protocol**

**8.5. Global Control Flow**

**8.6. Hardware Requirement**

**9. Algorithms and Data Structures**

**9.1. Algorithms**

**9.2. Data Structures**

**10. User Interface Design and Implementation**

**10.1. User Interface Design**

**10.2. User Interface Implementation**

**11. Testing**

**11.1. Unit Test Architecture and Strategy/Framework**

**11.2. Unit test definition, test data selection**

**11.3. System Test Specification**

**11.4. Test Reports per Spring**

**12. Project Management**

**12.1. 11.1 Project Plan**

**12.2. 11.2 Risk management**

**13. References**