**1. Project Vision**

1.1 Backgrounds: We are all IT/CS Students pursuing a Bachelor’s degree from Oakland University.

1.2

* Socio-economic impact: our project will have no socio-economic impact because it will never have a public release and offers no functionality not already offered by contemporary social media platforms.
* Business objectives: Our business’ objective is to pass Senior Capstone Project.
* Gap Analysis: Our application lends itself well to achieving our organization’s only objective.

1.3 Security and Ethical Concerns: Memcomb vows to not sell any user information submitted by its users, it has a strict Code of Conduct which bars its employees from using user information for inappropriate purposes.

1.4 Glossary of Key Terms

**2. Project Execution and Planning**

2.1 Team Information:

All members of the group added themselves into a Discord server created by Nathan. A Google Drive was created by Scott for the group to collaborate on the presentations. Github repository setup and maintained by team lead, David. Split into two focus groups: Front-end and Back-end, with assistance provided by both sides.

2.2 Tools and Technology: Memcomb makes use of Ionic, Angular, Node.js, and Firebase. Git is used in tandem with GitHub for version control. All members of the group are developing with the Windows operating system.

2.3 Project Plan

2.4 Best Standards and Practices:

Group members are encouraged to develop with Google’s Style Guide <http://google.github.io/styleguide/> to make code written by other people more easily understood.

**3. System Requirement Analysis**

3.1 Functional Requirements:

Our system shall

* Perform user registration
* Allow email verification
* Allow login of registered account
* Allow users to create posts
* Allow users to upload pictures and videos to their posts
* Allow users to become friends with other users
* Allow users to link related posts from other uses with posts of their own

3.2 Non-functional Requirements:

Our system should

* Securely store all user information
* Load pages within 2 seconds
* Upload files up to 400MB in size

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

3.4. Wireframe designs

**4. Functional Requirements Specification**

4.1. Stakeholders

* Our stakeholders will consist of those who find our app attractive

4.2. Actors and Goals

* Users want to be able to share, discuss, and view events in a social media style application with others.

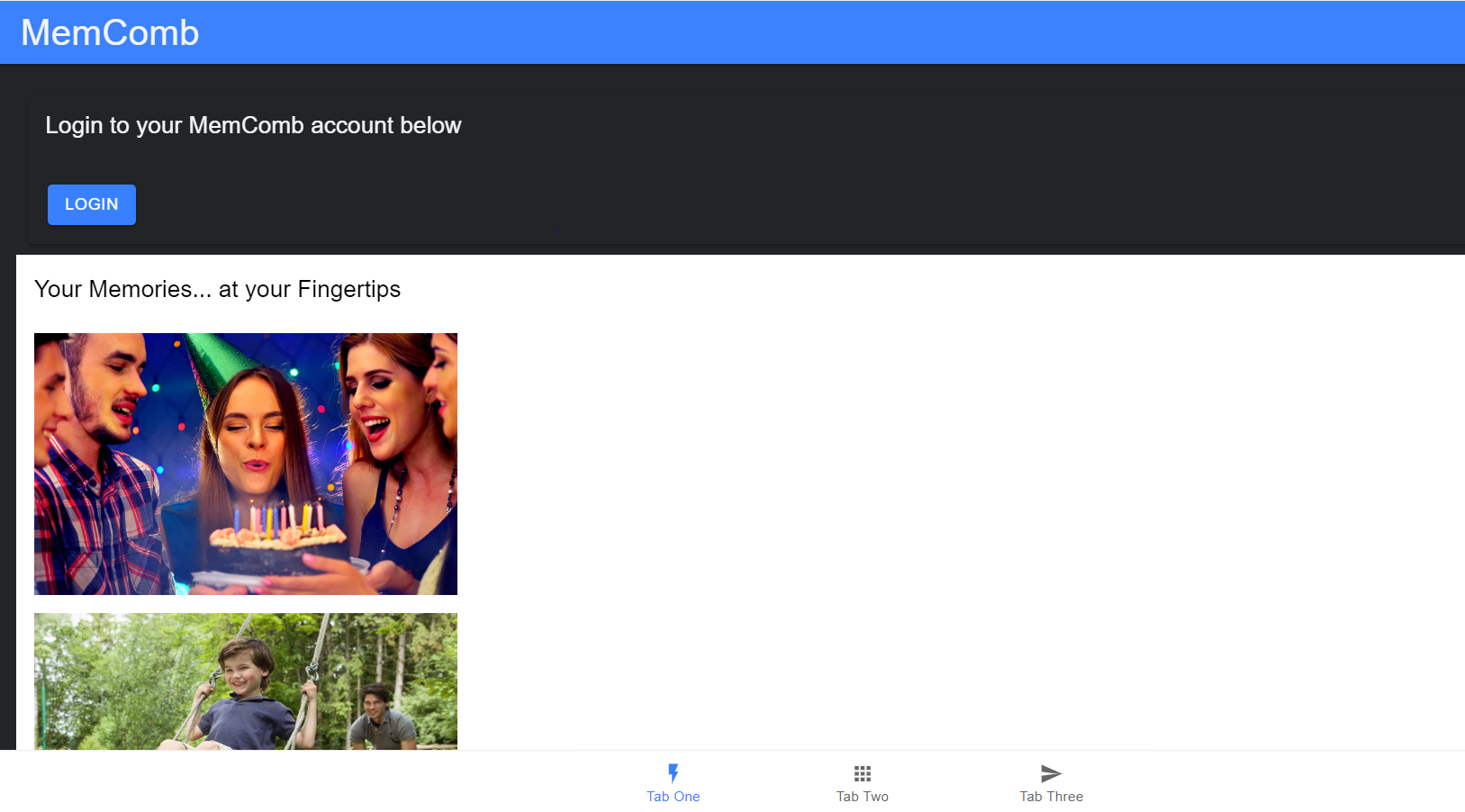
4.3. User stories, scenarios and Use Cases

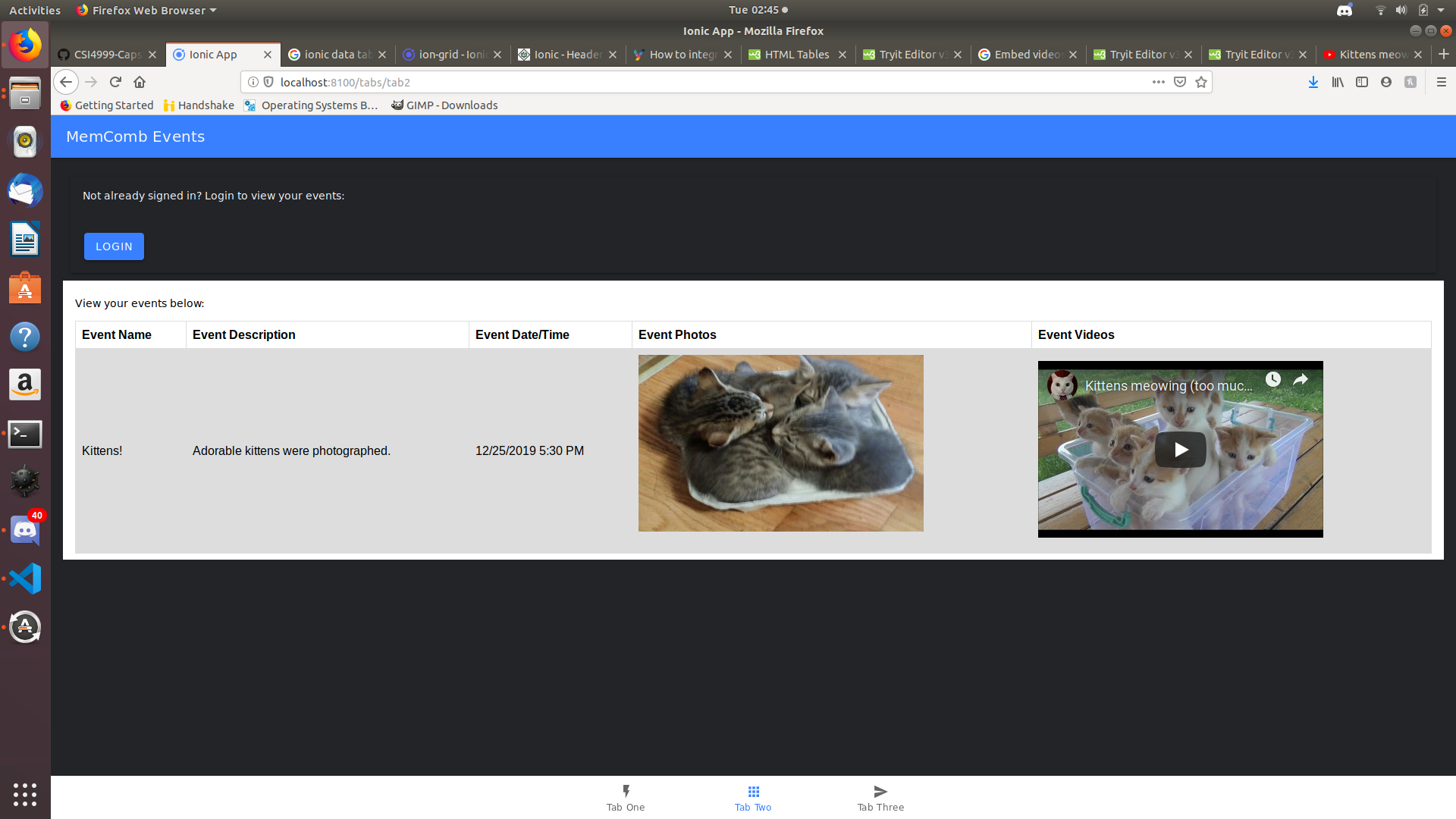
* Peter is going to a board game convention, and wants to share pictures of the board games he is trying with his friend Kyle, who is also attending the event. Peter wants Kyle to be able to view the pictures and note that he was at the event as well.
* Sam sees that Rachel posted a picture of a concert she went to. Sam attended the concert as well and wants to ask Rachel what her favorite band was. Sam wants to be able to comment on Rachel’s post and have her respond as well.
* Walder Frey is celebrating the marriage of his daughter, and wants to share pictures from this event. Frey wants to post before and after pictures of the celebration. Frey wants to be able to create the event and post multiple pictures throughout the event.
* Lucy went to a baseball game yesterday and posted pictures to her event. Lucy now wants to log back into the app and view her feed, checking on her post from yesterday. She wants to see if anyone she knew was also there, and if they posted additional pictures.

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**