MESA Everyday

MEMBERS:
CHRISTOPHER BARTLETT, MICHAEL COHOE,
FADI LABIB, MINWEI LUO, MINH NUGYEN,
THONG TRAN, MILLEN WAN

The Product

Gamifies the MESA experience for students (6th grade and up)

1. Collect core data about student experiences and their MESA journey
2. Give more concrete testimonials to obtain grant funding
3. Motivates students to participate in MESA sponsored activities and events to help them get college credits

Assumptions & Constraints

- Will be used by students as young as sixth grade
- Data integrity is questionable (student report – trust system)
- Adapt to different screen sizes
- Hosted using PSU domain
- Someone must maintain the codebase after us
- Using MESA colors, fonts, and pictures

Features Expected

- Landing page for registration and signing in
- Username and password recovery support
- Deactivating and changing account info support
- Responsive UI for the participants and the admins
- Participants can track and manage their progress
- Administrators are able to modify the rules of the game
- Google Calendar & countdown for major events

Features Actual

- Everything from Previous Slide +
- Users are able to see upcoming event specific to a badge
- Users & Admins get notification(s) for events in <= 2 days
- Users & Admins have access to calendar & countdowns
- Admins can change badge names and icons
- Admins can see top 3 scores for each badge

Deliverables

- User-side & Admin-side accessible in the World Wide Web
- Web app hosted on CAT infrastructure with SSL and HTTPS support
- Design and Testing Documentations
- Simple guide on how to use certain web app features
- Codebase for the web app minus sensitive keys and more obfuscation

Process Used

- Primarily uses the Waterfall model w/ some Agile elements
  1. Gather requirements from sponsor
  2. Design the product from the requirements
  3. Implement the product using designs
  4. Deliver the product
- Tested during implementation
- Daily Standup Meetings

Team Roles

Fadi Labib:	Project Manager, Risk Analyst, Product Owner
Michael Cohoe:	Developer, Architect: Backend, ~~QA~~
Thong Tran:	Developer, Architect: Frontend, ~~QA~~
Chris Bartlett:	Developer, Infrastructure & Hosting, ~~QA~~
Millen Wan:	Developer, ~~Security~~ Researcher, QA
Minwei Luo:	Developer, QA
Minh Nguyen:	Developer, ~~QA~~

Process & Schedule

Planned Process	Actual Process	Planned Schedule	Actual Schedule
Requirement Gathering	Defining MVP + stretch	Week 6 – Week 8	Week 6 – Week 8
Database Design	Fixing design/query issues	Week 6 – Fall Finals	Week 6 – Week 13
User's UI Design	Standardizing layout	Week 6 – Fall Finals	Week 6 – Week 17
Admin UI Design	Admin UI Design	Week 17 – Week 19	Week 17 – Week 19
Admin Research	Trial & Error -> Basic Approach	Week 6 – WB Week 4	Week 8 – Week 12
Prototyping	Adapting Prototype	Week 6 – Fall Finals	Week 7 – Week 13
Calendar /Events Design	Trial & Error -> Google Cal.	Week 6 – Fall Finals	Week 6 – Week 19
User-side Session Mgmt	User-side Session Mgmt	Week 10 – WB Week 2	Week 7 – Week 14
User-side Implem	User-side Implem	Fall Finals – Week 15	Fall Finals – Week 16
Admin-side Session Mgmt	Based on user Session Mgmt	Week 10 – WB Week 4	Fall Finals – Week 12
Admin-side Implem	Watered down Admin Implem	Week 11 – Week 16	Week 14 – Week 18
Automated Testing	Basic UT, mostly manual	Week 15 – Week 17	Week 14 – Week 16
Web-App Hosting	Beta and Production Hosting	Week 16 – Week 20	Week 17 – Week 20

Major Bugs & Challenges

- Calendar UI did not play nice
  - Considered pre-built calendar widget (end-of-life, fails with https)
  - Reverted to Google Calendar, without the horrible UI
- Database Design failed to account for rollback
  - Made major changes to all the queries in the program to support rollback
- Database timeout issue reduced app uptime availability to 8-hrs
  - Extended database interactive timeout, with scheduled mid-night resets

Major Bugs & Challenges

- Admin-side using Flask Admin doesn’t fit our use case
  - Used a hard-code admin role
  - Add some obfuscation in the code-base
    - Admin role is a secret word that is both hashed and salted
  - The Role column in the database can’t be modified
- Unit testing account management fails
  - Did an extensive manual test on it to insure that it works
  - Increased complexity and time for testing but successfully verified working

Lessons Learned

- Getting Frontend build experience
- Building an app with session management to authenticate users and administrators
- Working with each other and improving people skills
- Database sessions, rollback, commits, and uptime
- Using Google API and integrating widgets to a web app
- Debugging and Fixing client-server issues