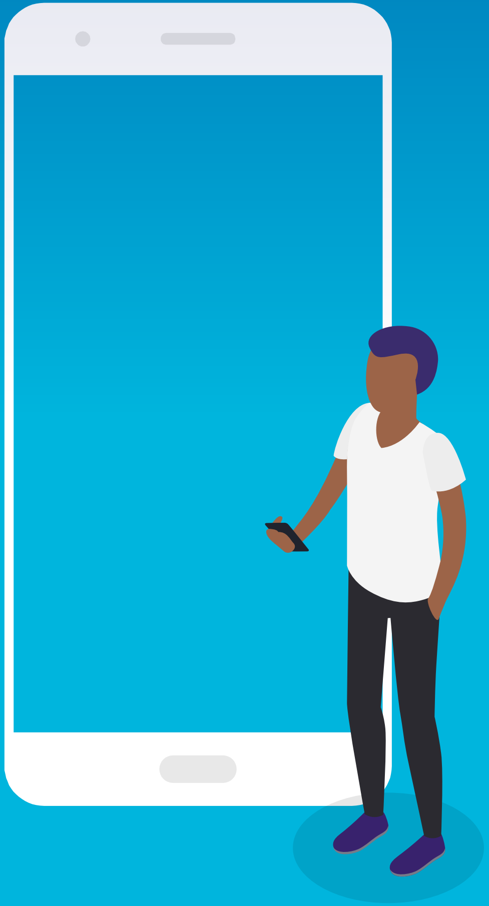
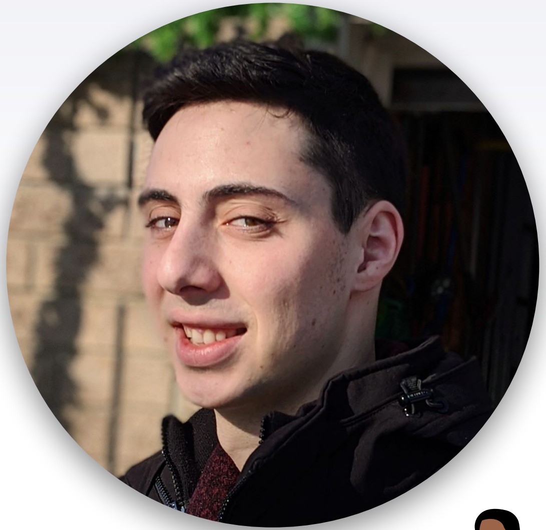
Class Application



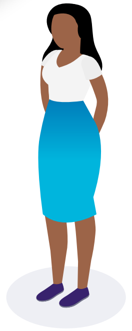
Our Team

Systems Developer
Database Admin

Process Analyst
Quality Assurance Analyst
Designer



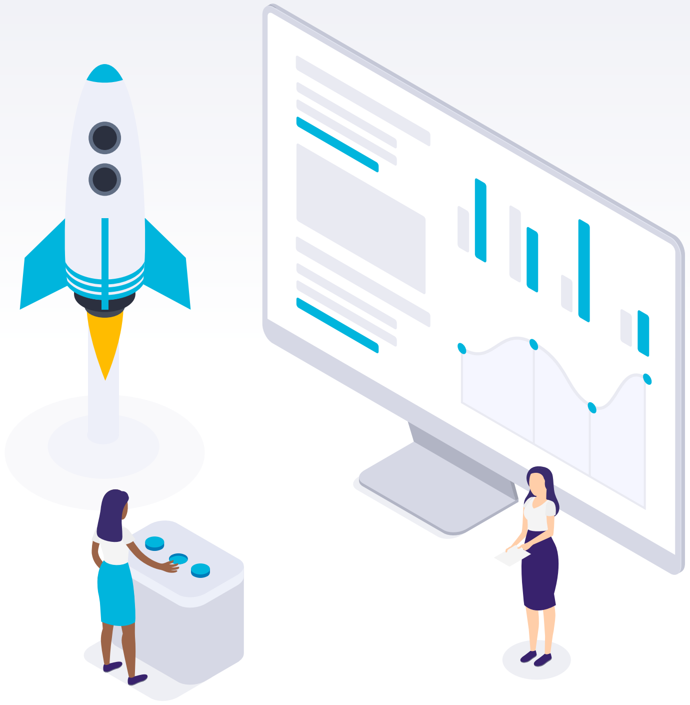
Engineer
Team Lead
DevOps



1

Introduction

Let's start with our description and problem statement



Proposal Timeline

- Outline Project **Objectives**
- Define **Measures of Success**
- Estimate Project Duration with **Activity Diagram**
- Perform **Risk Analysis** and note **Countermeasures**

Project Goals

- Develop centralized platform where students & alumni get updates about courses & events
- Helps management measure performance of school
- Develop Mobile and Web Applications Simultaneously

Objectives

- Application will be modularized based on user groups (student, management & alumni)
- Use **Banner** for authenticating all user groups.
- Web and Mobile platforms will use similar tech stack
- Program has Messaging System for communication between all users
- Management has extra powers:
  - Add/drop classes
  - Track discussions & messaging for harassment

Measures of Success

- Separate modules for each user group, with its own features (moving users between groups is also possible)
- Program redirects to **Banner** for logging in users
- Simplified maintenance due to similar tech and tools used by program
- Intra- and Inter-group communication is successful (and chat-room creation is enabled)
- Program informs management of harassment and cyber-violence
- Management is able to add/drop and edit classes

Technology Stack

Frontend

- React and React Native
  - Can use same framework for mobile and web

Server

- Go-Lang for HTTP Server with in-built concurrency
- Will work as an API with RESTful API endpoints or GraphQL
- NGINX for reverse proxy

Database

- Relational and user-permission managed
- Will handle events such as updates, harassment tracking, ...

Deployment & Production

- Amazon Web Services for hardware management
- Combine with Docker for CI deployment pipeline

Security

- Handled by the university
- SSL Encryption with Bearer-Token access

Testing

- Continuous Integration with **TravisCI**
- UAT Testing for mobile and web separately

Project Estimation

COCOMO

- Will follow **Semi-Detached** system model

- $E = a\,(KLOC)^b$
  $= 3.0\,(15)^{1.12}$
  $= 62.28$ person-months

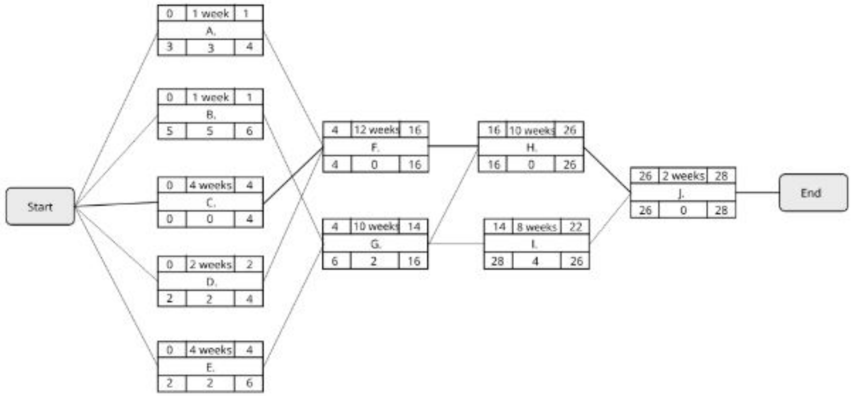
Albrecht Complexity Multipliers

- Used to estimate total function points

External User Types	Estimate	Complexity	Total
EI	6	4 (Medium)	24
EO	10	5 (Medium)	50
EQ	4	3 (Low)	12
LIF	4	10 (Medium)	40
EIF	2	10 (High)	20
		Total Function Points (FP)	146

Activity Diagram

- A: Development Server Setup (1 wk)
- B: Mobile Env Setup (1wk)
- C: Data collection (4wks)
- D: Create software architecture for server (2wks)
- E: Design UI/UX (3wks)
- F: Server Development (6 sprints, 12wks)
- G: Mobile application Development (5 sprints, 10wks)
- H: Quality Assurance (5 sprints, 10wks)
- I: User Acceptance Testing (2 sprints, 2wks)
- J: Set up production server and deployment (1 sprint 2wk)



Risk Analysis

Tech Setup Complications

- Initial setup may be hard to modify in later stages
- **Solution**: Conduct extensive prior research

Vague Requirements

- Requirements spec. page is too broad
- **Solution**: Follow Agile and connect with client after every increment

Inconsistent Data

- Test data from client can have faults.
- **Solution**: Spend time cleaning and pre-processing data

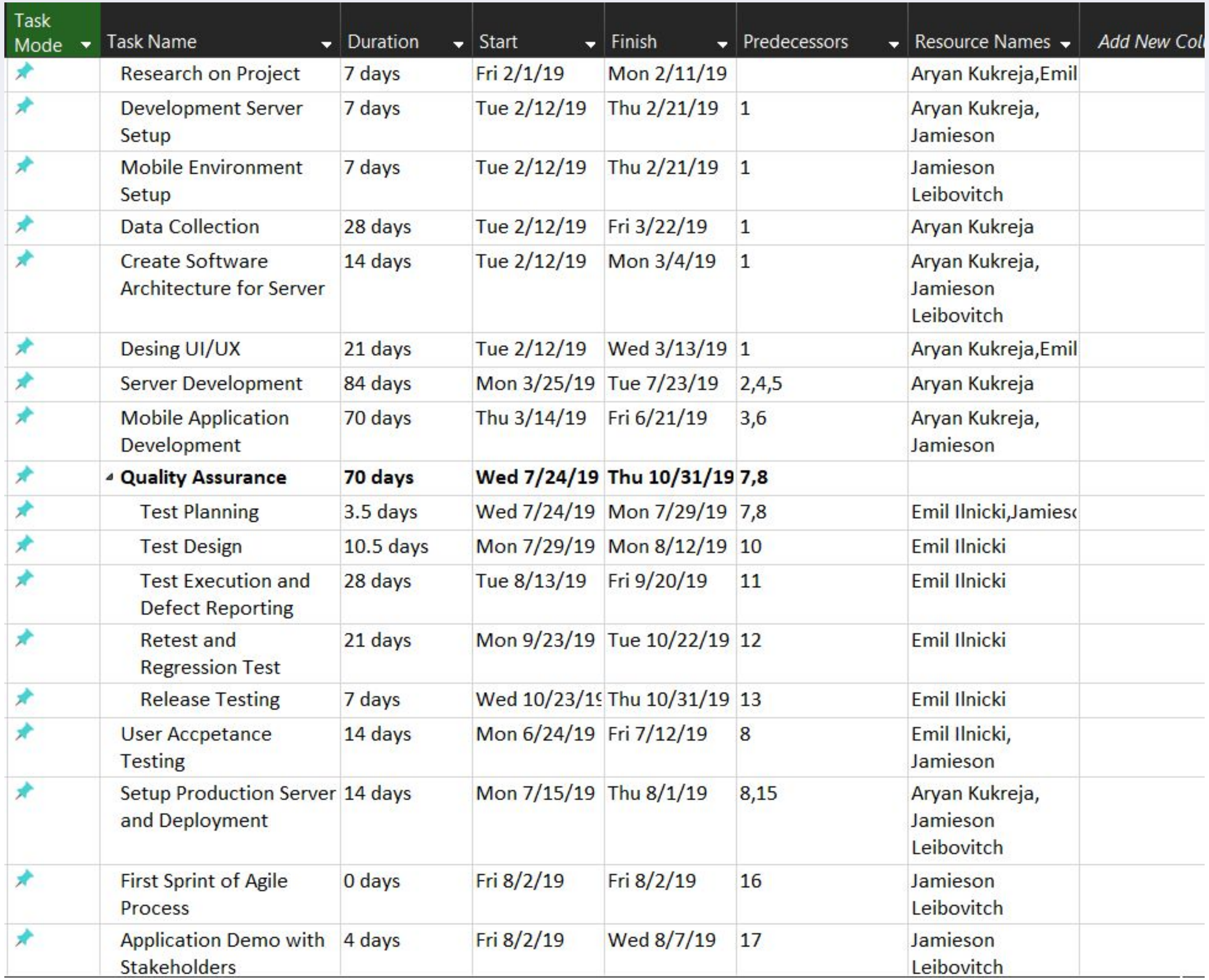
Unclear Client Design

- Requirements can be misinterpreted
- **Solution**: Use SQA group to interface with client

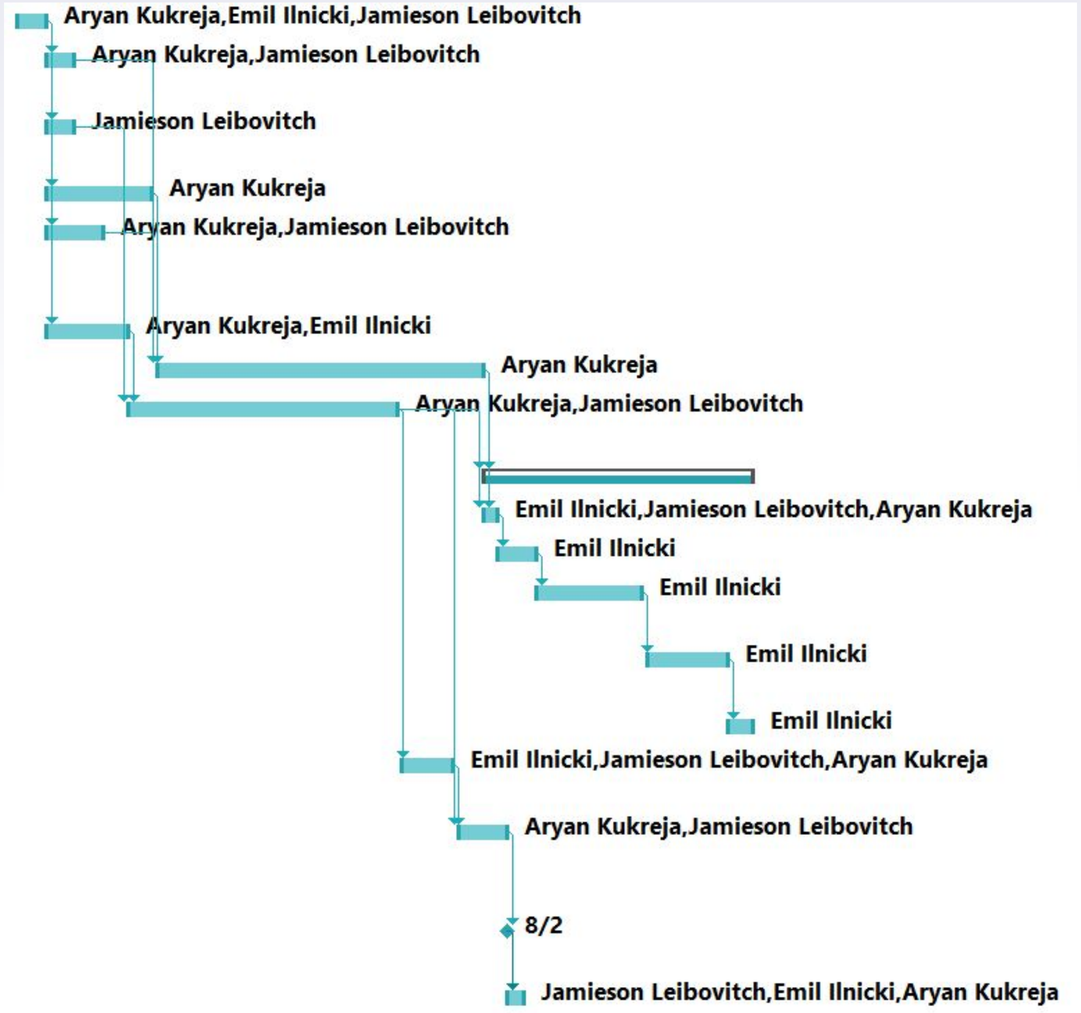
Missing Sprint Deadline

- Increment deadlines missed in Agile process can delay final product
- **Solution**: Reduce increment size; *under-estimate* expectations

Gantt Chart



Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names	Add New Col
📌	Research on Project	7 days	Fri 2/1/19	Mon 2/11/19		Aryan Kukreja,Emil	
📌	Development Server Setup	7 days	Tue 2/12/19	Thu 2/21/19	1	Aryan Kukreja, Jamieson	
📌	Mobile Environment Setup	7 days	Tue 2/12/19	Thu 2/21/19	1	Jamieson Leibovitch	
📌	Data Collection	28 days	Tue 2/12/19	Fri 3/22/19	1	Aryan Kukreja	
📌	Create Software Architecture for Server	14 days	Tue 2/12/19	Mon 3/4/19	1	Aryan Kukreja, Jamieson Leibovitch	
📌	Desing UI/UX	21 days	Tue 2/12/19	Wed 3/13/19	1	Aryan Kukreja,Emil	
📌	Server Development	84 days	Mon 3/25/19	Tue 7/23/19	2,4,5	Aryan Kukreja	
📌	Mobile Application Development	70 days	Thu 3/14/19	Fri 6/21/19	3,6	Aryan Kukreja, Jamieson	
📌	**Quality Assurance**	**70 days**	**Wed 7/24/19**	**Thu 10/31/19**	**7,8**		
📌	Test Planning	3.5 days	Wed 7/24/19	Mon 7/29/19	7,8	Emil Ilnicki,Jamies	
📌	Test Design	10.5 days	Mon 7/29/19	Mon 8/12/19	10	Emil Ilnicki	
📌	Test Execution and Defect Reporting	28 days	Tue 8/13/19	Fri 9/20/19	11	Emil Ilnicki	
📌	Retest and Regression Test	21 days	Mon 9/23/19	Tue 10/22/19	12	Emil Ilnicki	
📌	Release Testing	7 days	Wed 10/23/1	Thu 10/31/19	13	Emil Ilnicki	
📌	User Accpetance Testing	14 days	Mon 6/24/19	Fri 7/12/19	8	Emil Ilnicki, Jamieson	
📌	Setup Production Server and Deployment	14 days	Mon 7/15/19	Thu 8/1/19	8,15	Aryan Kukreja, Jamieson Leibovitch	
📌	First Sprint of Agile Process	0 days	Fri 8/2/19	Fri 8/2/19	16	Jamieson Leibovitch	
📌	Application Demo with Stakeholders	4 days	Fri 8/2/19	Wed 8/7/19	17	Jamieson Leibovitch	


Aryan Kukreja,Emil Ilnicki,Jamieson Leibovitch
Aryan Kukreja,Jamieson Leibovitch
Jamieson Leibovitch
Aryan Kukreja
Aryan Kukreja,Jamieson Leibovitch
Aryan Kukreja,Emil Ilnicki
Aryan Kukreja
Aryan Kukreja,Jamieson Leibovitch
Emil Ilnicki,Jamieson Leibovitch,Aryan Kukreja
Emil Ilnicki
Emil Ilnicki
Emil Ilnicki
Emil Ilnicki
Emil Ilnicki,Jamieson Leibovitch,Aryan Kukreja
Aryan Kukreja,Jamieson Leibovitch
8/2
Jamieson Leibovitch,Emil Ilnicki,Aryan Kukreja

THANKS!

Any questions?