



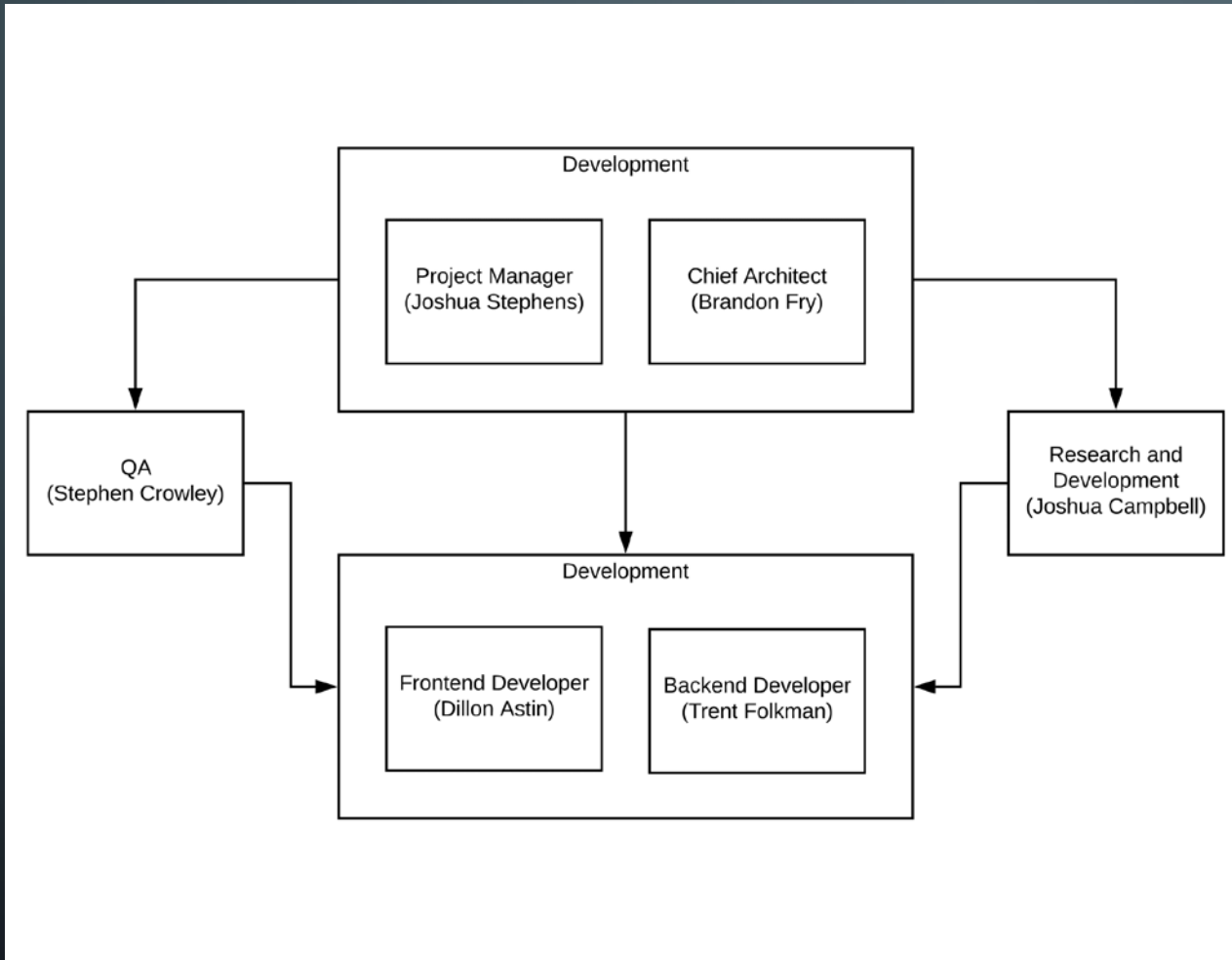
CS 428 – ORG CHART REVIEW

FALL 2018 – BRUCE F. WEBSTER – 10/1/18

REMINDER: WHY HAVE AN ORGANIZATION CHART?

- Division of labor
- Focus talents and interests
- Make responsibility clear
- Establish lines of communications that reflect the target architecture (see [“Remember Conway’s Law”](#))
- Align individual goals with team goals (see [“How to retain IT talent with goal alignment”](#))

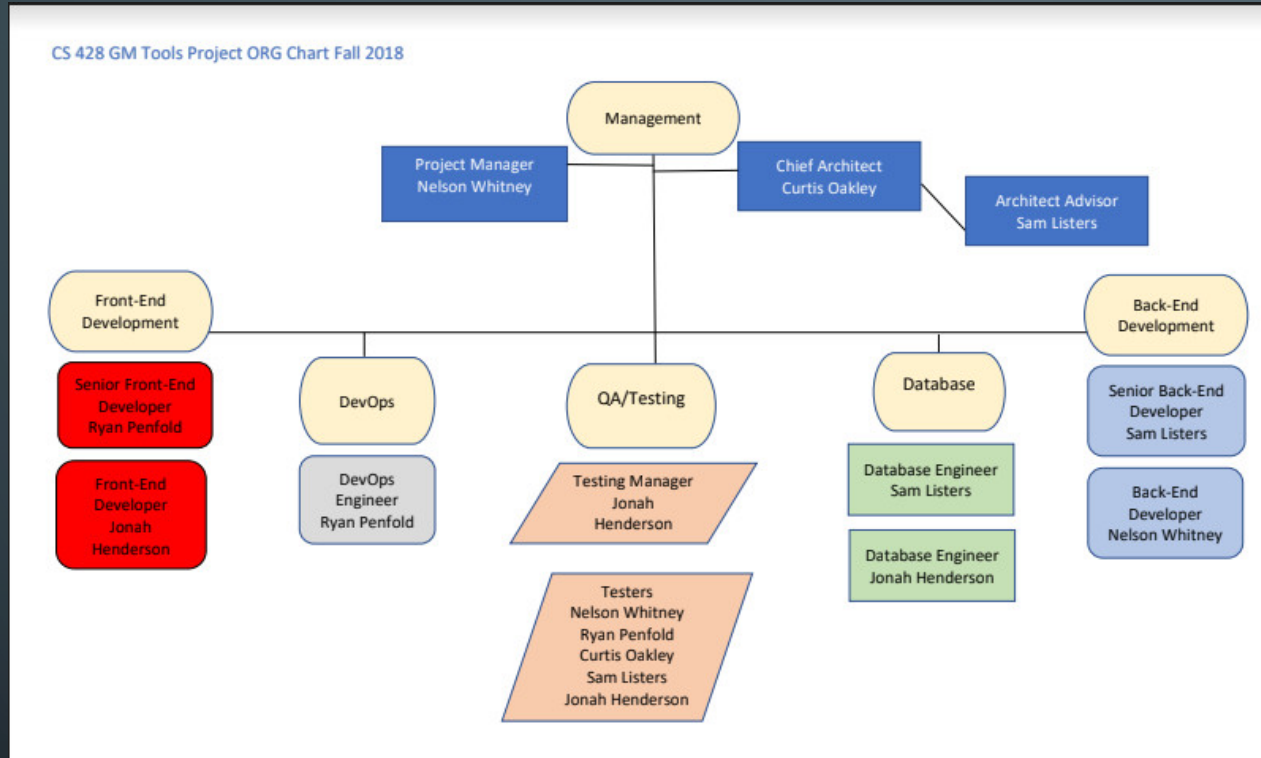
PYTHON GAMES



Use Python to rebuild an already existing game. Start with Mario level 1 to see how easy or difficult creating a game would be. If creating this level seems too easy, we would expand by creating more levels of Mario, or even create other games such as chutes and ladders.

RISK: Creating a game is much more difficult than we thought. Pygames is not very intuitive.

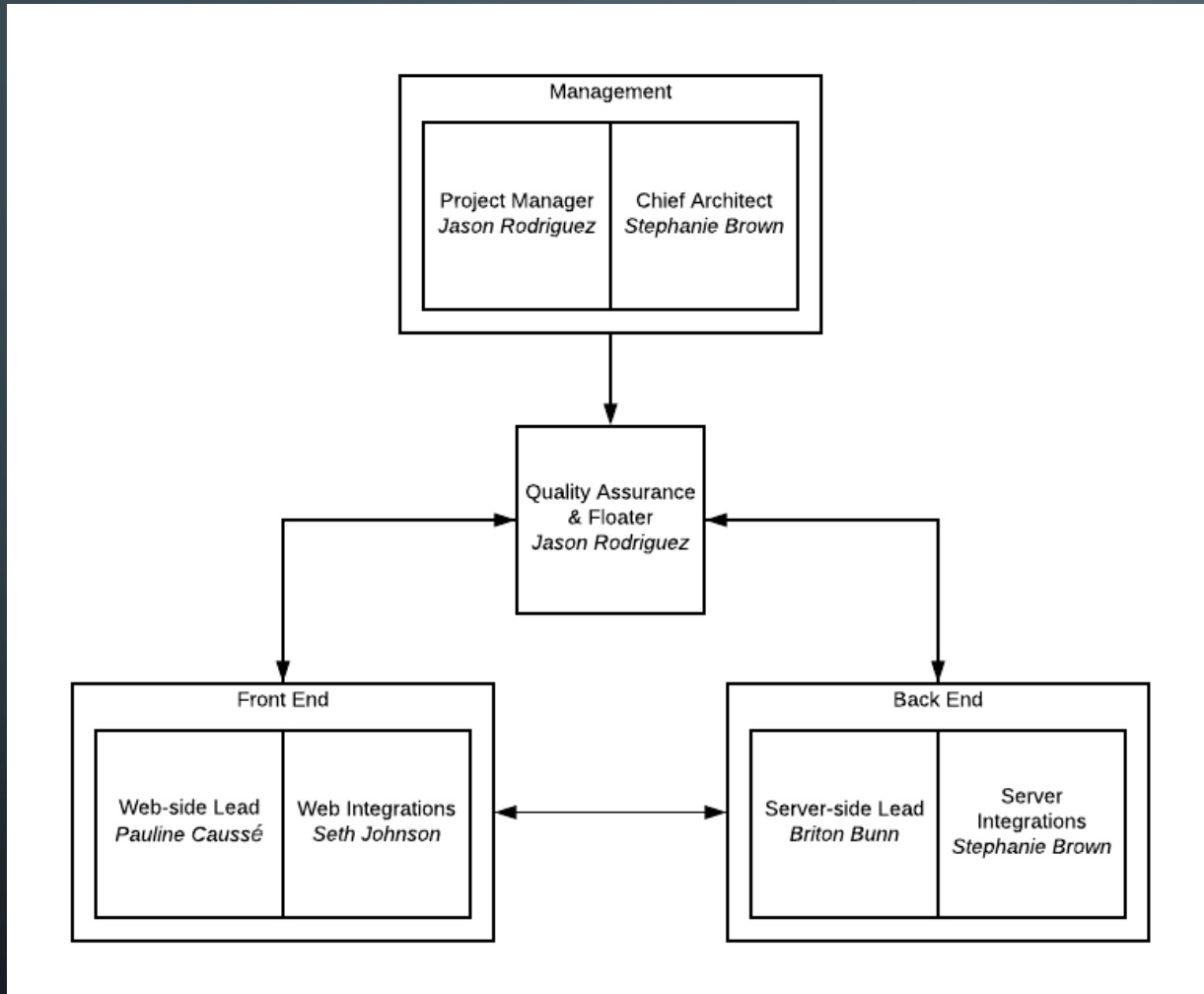
GM TOOLS



This project is designed to implement many of the tools needed by Game Masters in most tabletop games. It will include key features such as dice rolling, initiative trackers, loot drops, item/monster lookups as well as additional features such as GM Note taking. This will all be downloadable by the GM as well as the players of the game. Even if the players are offline, they should still be able to connect to each other. When players are online, the data will then be backed up to the server.

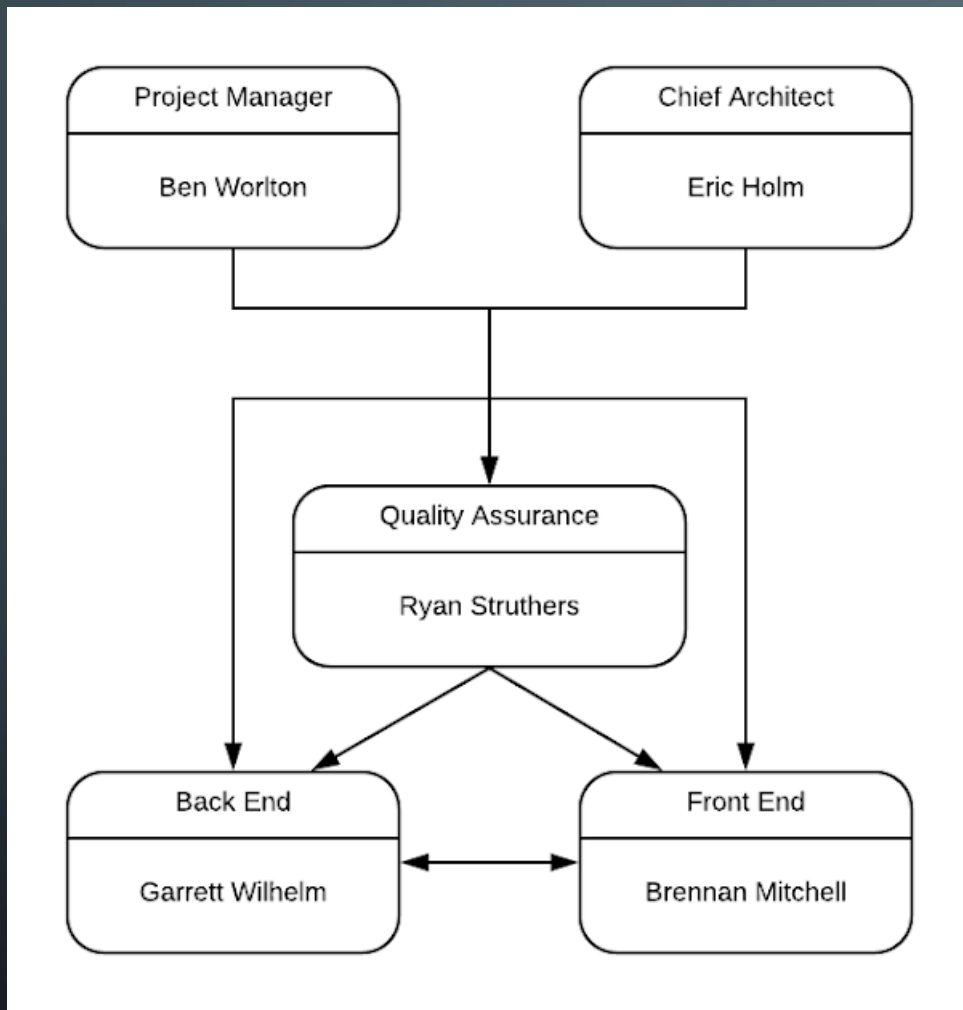
RISK: We have a lot to accomplish. It is possible we may not be able to implement all the ideas we have had for this project.

CLC PAINTING



Our project consists of 5 team members filling the roles of Chief Architect, Project Manager, Server-side Lead, Server-side Integration, Web-side Lead, Web-side Integration, and Quality Assurance. The front-end team, which consists of web-side lead and integration will be responsible for all things user-facing. The back-end team will be responsible for implementing a server with APIs accessible to the front-end as well as databases for all object data. The QA will be responsible for assuring quality and correctness of product functionality.

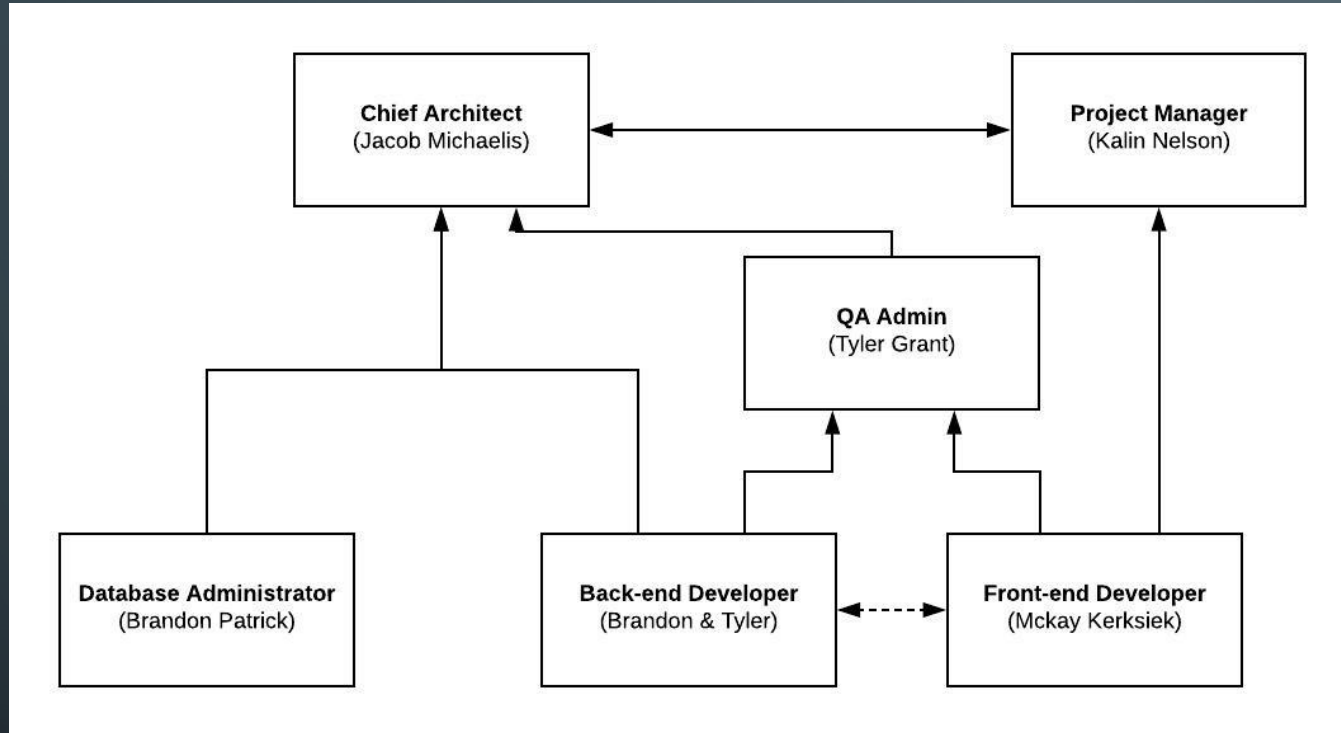
WACKY COMPETE-FOR-2ND GAME



Fashionably Late is a multiplayer party game, wherein everyone's goal is to fight to arrive second place at the premiere party of Francis Vaughn Burgoise.

Thus far, we have done the initial setup work for the project, including setting up our repository, defining/assigning roles, and creating an organizational chart. We have already continued onto the next steps of getting more familiar with our tools (Unity and C#), and establishing our own ideas for how we'd like the game to play out. Our next meeting will be an opportunity to bring together our different thoughts and establish the core gameplay/functionality of our game.

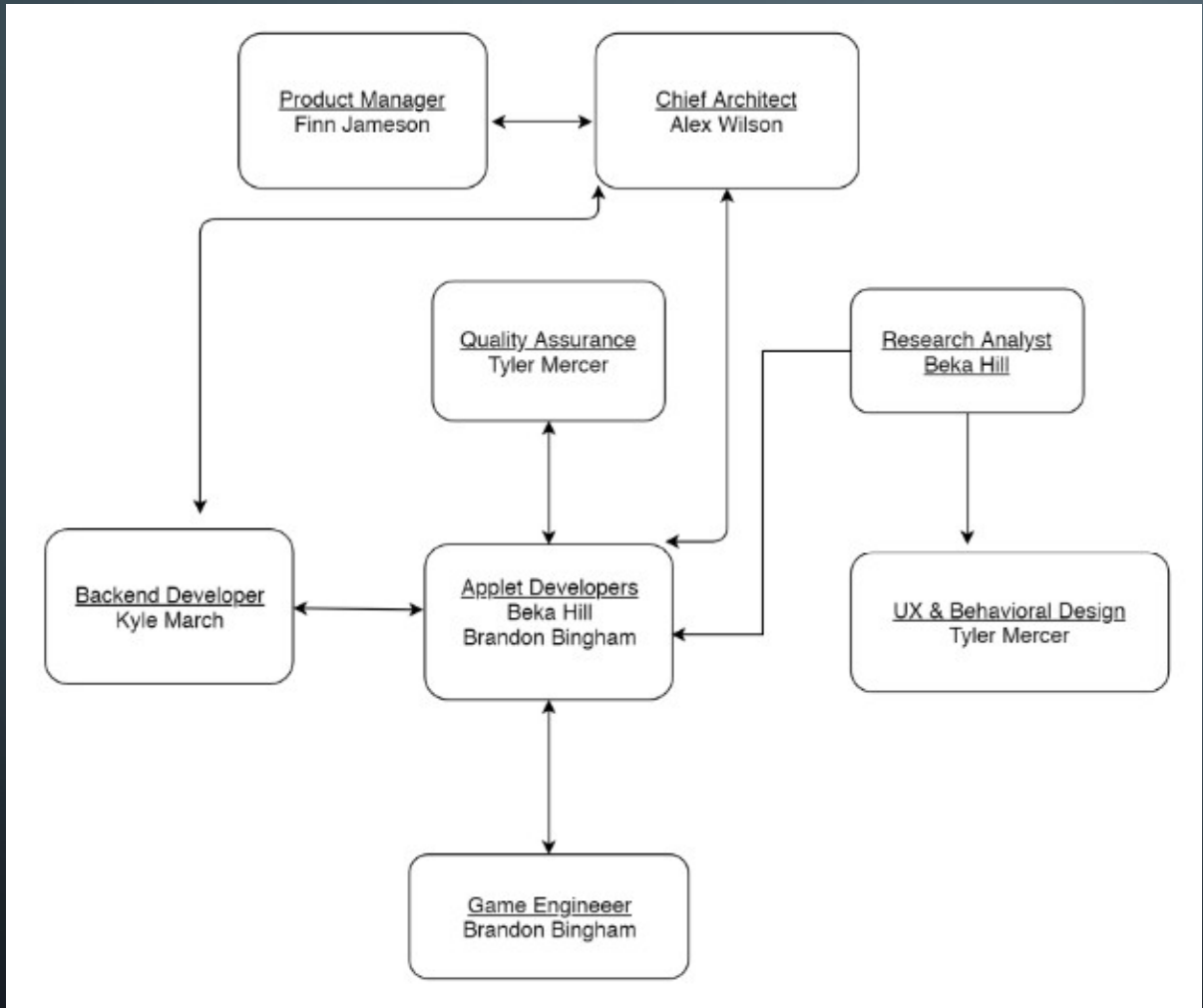
LIFE TRACKER



We are making an Android app that will help the user keep track of life details. We narrowed it down to finances and maintenance tracking this week. It will keep track of purchases that the user makes and allow them to keep track of maintenance things (car maintenance, house maintenance, etc) by allowing them to record when the last time they did something was, and set reminders for when their next thing should be done.

RISK: Finding times where all of us can meet is difficult for our team, so it is kind of a struggle to get things done together.

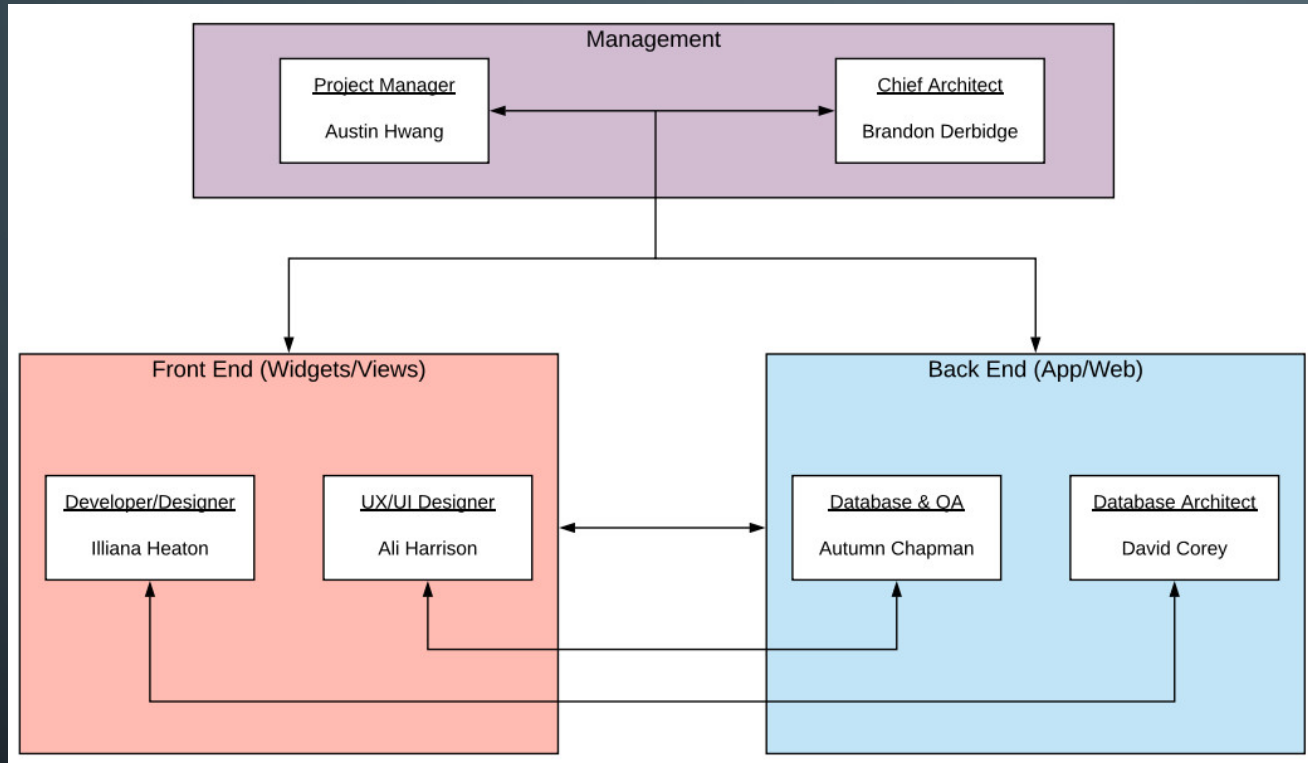
MATH BUT MORE



We are attempting to build a website that encourages middle schoolers to enjoy solving math problems. We are starting this by creating applets that provide examples of the potential of the concepts they are learning have.

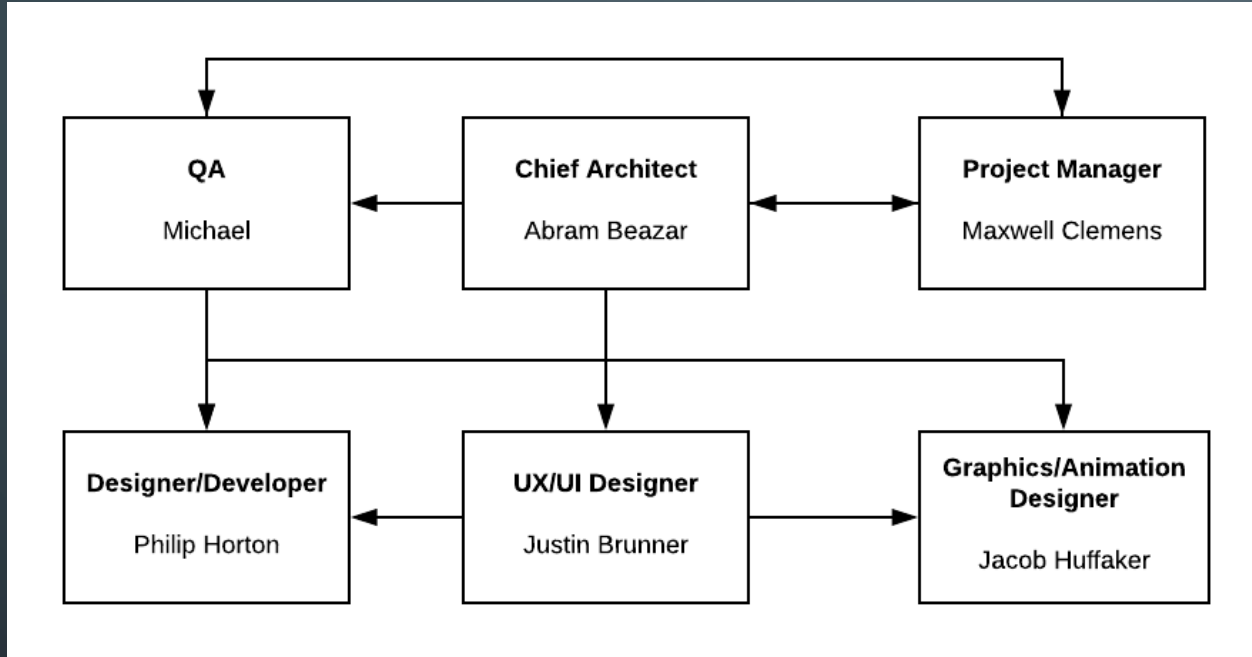
Risk: We currently are still figuring out the concreteness of our roles and what we want the site to [do. Our] fear is that we end up not deciding and agreeing on what to do.

PROP OWNER



To produce a usable Housing Management application for vendors, owners, and tenants. Focus will be placed on primarily the vendor application side, in which several views such as the vendor profile creation page, request history, profile view, and owner vendor search views will be generated and connected to a central database. Firebase will be the database management storage, and flutter the framework that will contain dart code for multi-platform application creation.

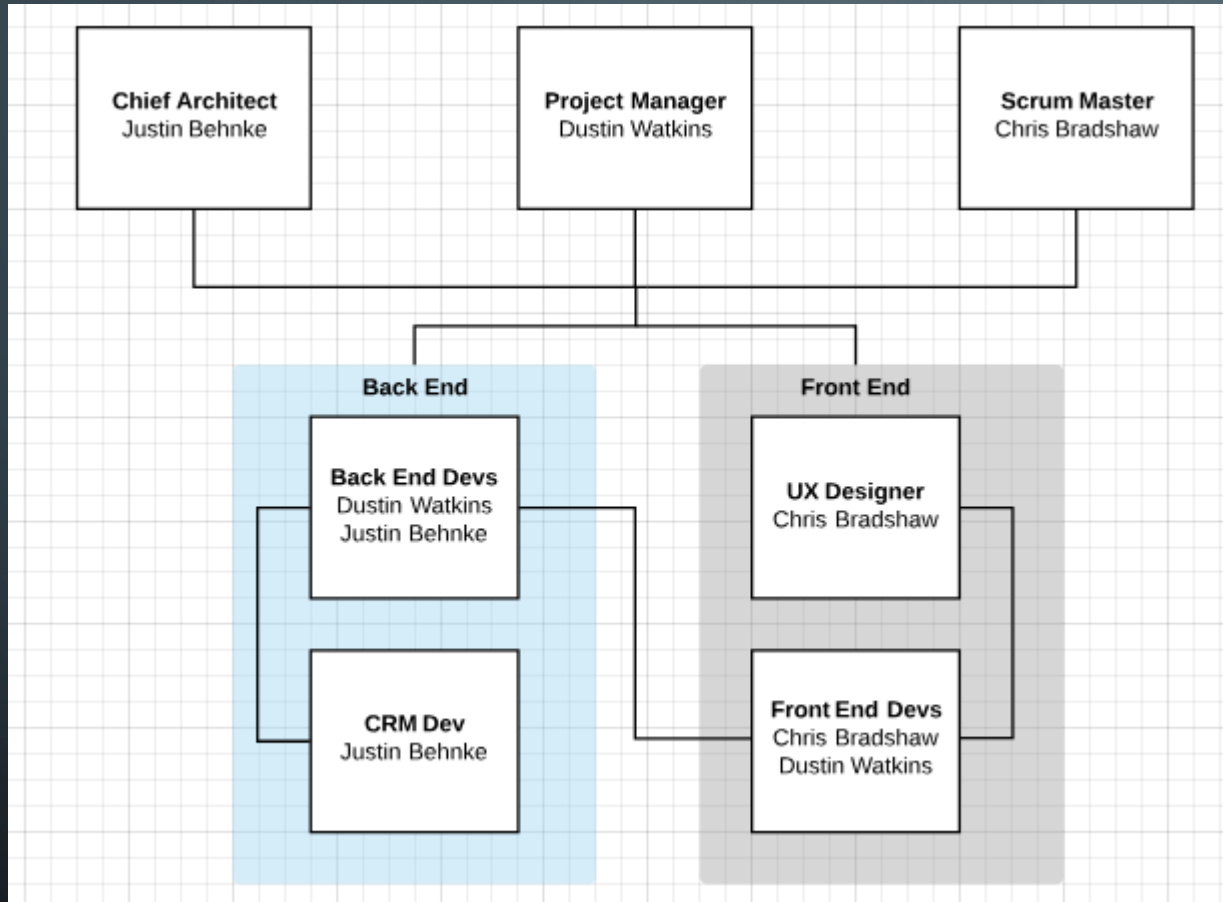
GAME: PUZZLE RPG



A turn-based RPG where every battle is designed to be won in a certain creative way.

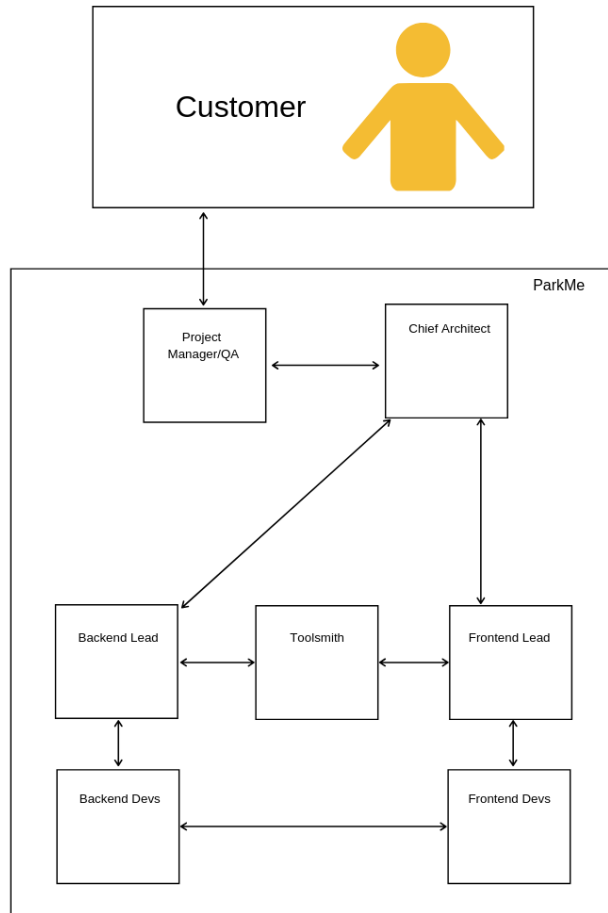
TASKS: Learn Unity, Create game framework, Create sprites, Design battles, make integration tests.

DATA ANALYTICS TOOL



Web App that allows a sales organization who uses Salesforce to identify where talent development in their sales pipeline would most greatly increase revenue.

PARK ME!

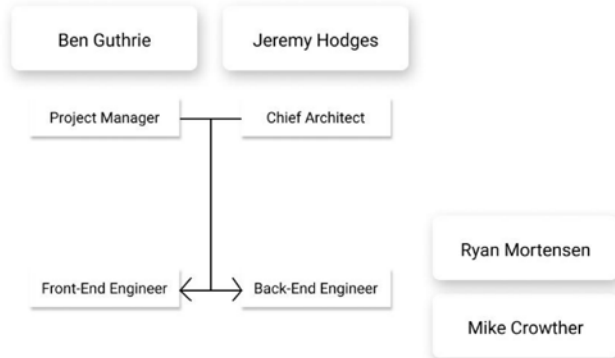


We have created our project repository, given everyone responsibilities and roles, and created an org chart, and begun to define what specific problem within the parking spot sharing industry that we can solve. Because our team is one of the larger ones, we are trying to prepare a well-defined project specification to eliminate complexity within our project. In the next week we will create an architecture model for the builders and create some user stories to outline the experience of interacting with our app

RISK: Defining our user stories and exact problem set will need to be clear and detailed for maximum benefit.

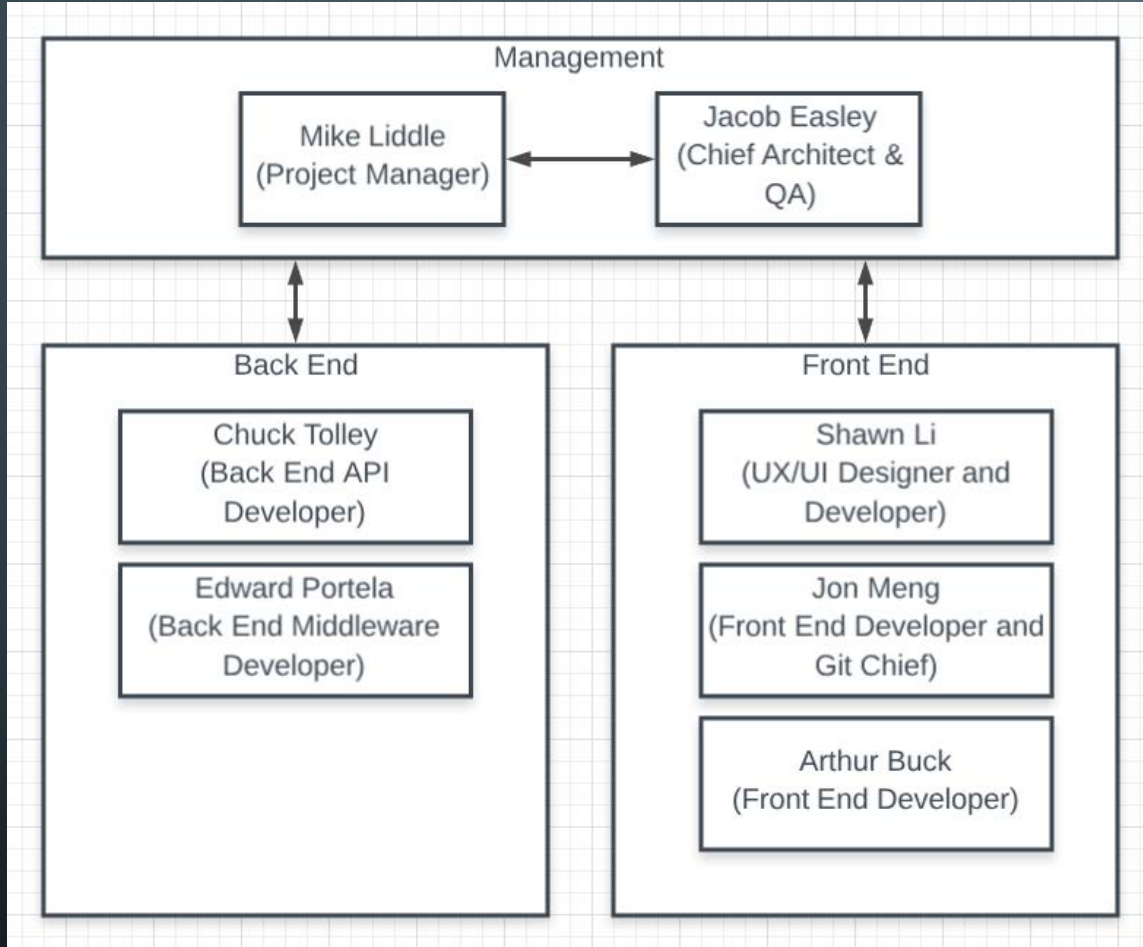
SOCIAL MEDIA FOR FAMILY EVENTS

(status report was empty)



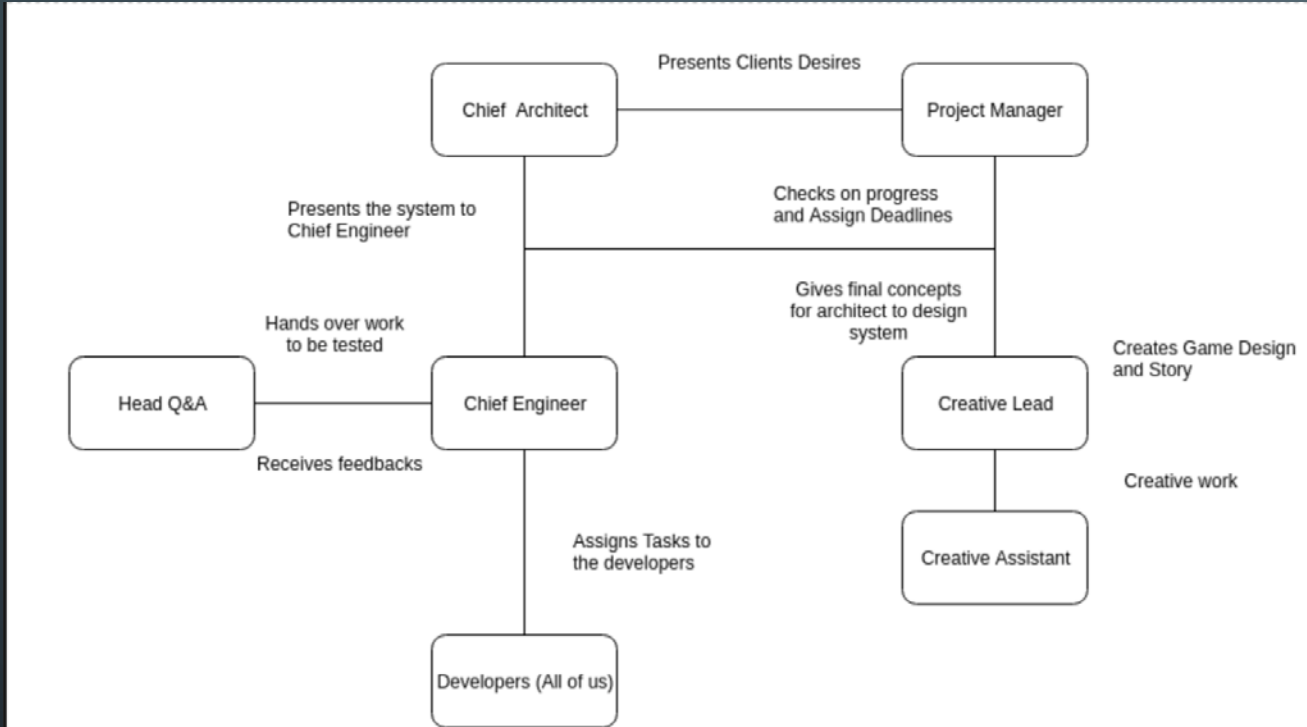
Ryan Mortensen	Jeremy Hodges	Ben Guthrie	Mike Crowther
<p>Back-End Engineer</p> <p>Responsible for creation, maintenance, and testing of web server and services</p> <p>Responsible for interactions with Family Search API</p>	<p>Chief Architect</p> <p>Front-End Engineer</p> <p>Final word on architectural decisions such as software design patterns</p> <p>Responsible for design, implementation, and testing of front end interface</p>	<p>Project Manager</p> <p>Front-End Engineer</p> <p>Responsible for managing team task board, submitting status reports, and overseeing testing</p> <p>Responsible for design, implementation, and testing of front end interface</p>	<p>Back-End Engineer</p> <p>Responsible for creation, maintenance, and testing of web server and services</p> <p>Responsible for interactions with Family Search API</p>

TRASH TALK



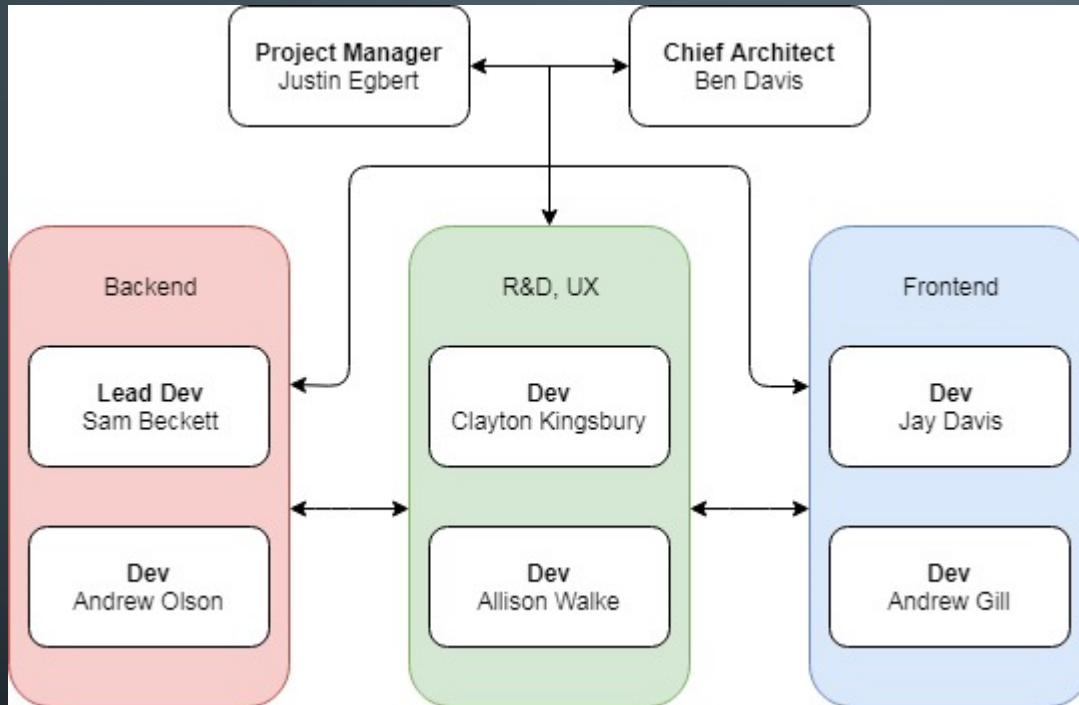
Build a web portal to allow customers of TrashTalk to interact with their data. The web interface will be mobile-friendly, having a clean, modern look with a minimal amount of features. A separate API will be provided for publishing and consuming events and updating data as needed.

TEXT-BASED ADVENTURE GAME



Create a text adventure game similar to Zork using Java. Currently made to be single player, but may be multiplayer

PILE: A SELF-ORGANIZING NOTES APP



Pile is a self-organizing notes app.

RISKS:

- React might be harder than we think.
- Fewer of us know react than we hoped, but we all want to learn.