

ARMR's Team Plan

Tasks

[Calendar of intermediate and final milestones for tasks](#)

Minimum viable product

[Identification of minimum viable product for first release](#)

[Issues postponed \(eg. security mitigations, user interface elements\)](#)

[Value to users](#)

[Provides opportunity for feedback](#)

Tasks

List of tasks, expected effort, allocation to team members

Task Name	Description	Estimated Time	Started for MVP	Assignee
Twillo Intergration	Integrating twillo into the application, sending initial invites as well as final reminders	3 hours		Ryan
Homepage React	Creating react components for homepage	3 hours	✓	Anna and Maddie
Login page React	Creating the login page	2 hours	✓	Anna
New party react components	Creating the new party page	3 hours	✓	Rachel
Schema Creation	Creating the schema to allow future progress	2 hours	✓	Full team

Login Functionality	Creating login functionality (use passport.js?)	2 hours	✓	Maddie
Model functions	Creating basic queries	3 hours	✓	Rachel
CSS/Improving UI	Making the app visually pleasing both on desktop and mobile	5 hours?		Full team
Testing model functions	Writing unit tests for model functions	3 hours	✓	Maddie
Routes	Creating RESTful routes	4 hours?	✓	Full team
Security	Implementing security mitigations	4 hours		Ryan
Cost-sharing	Adding cost-sharing functionality	4 hours		Full team

Calendar of intermediate and final milestones for tasks

Sun	Mon	Tues	Weds	Thurs	Fri	Sat
13:	14:	15:	16: + create basic hello world react app	17: + start login page react + start homepage react +start new party react	18: FIRST MEETING WITH MENTOR, Go over basic outline of MVP + create schemas	19
20: + finish react components + begin model functions	21: + finish model functions + routes! + model tests	22: + routes ! +login functionality	23: MVP DUE AT NOON	24: TURKEY DAY	25: break	26: Break

27: break	28: PEER REVIEWS + revise design + begin twillow implement ation + cost splitting functionalit y	29: + PROPOSE D 2nd MEETING WITH MENTOR	30: REVISED DESIGN DUE +finish twillow and cost splitting	1 + begin css + begin debug + security measures	2	3
4	5 + PROPOSE D THIRD MEETING	6 + finish css + doub	7: DEMO TO MENTOR	8 + revise code with comments from mentor	9	10
11	12: FINAL CODE HAND IN / PRESENT ATION	13:				

Minimum viable product

Identification of minimum viable product for first release

The MVP for our project includes all of the basic functionality of potluck besides notification and cost-sharing. The user will be able to invite friends to be a part of a “party” and they can sign up for items to bring to the party. The user interface will be implemented to the point where all functionality on the users side is available.

Subset of concepts to be included

1. Party

- User able to create party and identify guests, location, time, description, and supplies upon creation
- Host able to edit guest list
- Only users invited to party should be able to view the party’s information

2. Item
 - Attendees able to add items for a party
 - Guests able to sign up for unclaimed items
3. Supply list
 - Each party has a supply list
 - Items able to be added to supply list
 - Items able to be claimed by users on supply list

Issues postponed (eg, security mitigations, user interface elements)

In the MVP, we will not concern ourselves with Twilio (the api that we will use to send text messages). Due to not paying attention to this, we will not implement the reminders a few hours before a party, or the initial text inviting them to the party.

We will also leave the cost-sharing functionality of Contributions until after the MVP, because this could involve an external API (such as PayPal). Since it is not part of the core functionality of our app, it appears to be time-consuming, and we can still have guests sign up to bring items without implementing cost-sharing, we will save this for the final version.

In addition, we will leave the UI tweaking until the full version. As mentioned above, the UI will be functional, but may not look pretty or fully fleshed out in the MVP. We will concern ourselves more with the functionality and making sure that each individual part of the application works before fussing with the UI.

Finally, we will postpone security mitigations as thinking through all the different risks may take a lot of time, and our design may change while we are in the course of creating the MVP. Once we have the application all set, then we will go about mitigating the risks.

Value to users

Our MVP provides value to users because a user can better organize, see, and therefore know the guests and supplies of a party. Our MVP displays a party's location, description, time, guests, and supplies all in one place. Users can update the supply list and see who is bringing what. Our MVP will also allow the users to create an invite list for a party and decide who is coming. It will additionally make sure that only those invited will be able to view the details of a party.

Provides opportunity for feedback

Our MVP provides opportunity for feedback because it implements all of the major functionality of our application besides notifications and cost-sharing. It also provides our peer reviewers with the core components of our application's codebase such as the express.js file, routes, react components, and login functionality. The final steps to finishing our application, namely integrating with the Twillo API, adding cost-sharing functionality, improving UI, and adding more complex security measures, can easily be added on top of this core codebase.