REP Los Angeles

1. Vision Statement / Mission:

Our food application will show you all the street food vendors closest to you! It will focus on elusive vendors that sell a variety of foods. Eventually, we would like to expand to other street vendors such as taco stands, hot dog carts and other specialty food stands

There will be separate accounts for vendors and for customers. There is one mobile application where both the user and vendor can log onto. It will default as a user login, but allow for vendor sign in on the same page.

Users will have a map view of their location with the nearest vendors. In map view, each vendor will show up as an icon indicating their vendor type (food trucks will have a truck icon, food stands will have a food stand icon, etc.).

Functionalities:

- Consumer Application
    - Create/Edit/Delete Consumer Account
    - View closest food vendors (geo-tracker)
    - Filter/Sort vendors by different categories
    - Post photos (scan for inappropriate images, prevent posting if necessary)

- Vendor Application
    - Create/Edit/Delete Vendor Account
    - Must be able to check legitimacy of vendor
    - Set schedule (Fixed or Live)
    - Set up location

Functional Requirements:
- Create Accounts
- Location Mapping
- Update Information (Vendor and Customer)
- Filter by Distance, and/or Type
- Go Live or Set Schedule (Vendors)

Non-Functional Requirements:
- Database

- Implement location mapping
- Security (Validate Vendors)
- Check for existing accounts
- Intuitive UI design

2. Roadmap:

Development Languages: Javascript, JSX

Deployment Environment: iOS, Android

Deployment Characteristics:

3. Development Methodologies:

User Cases:

Use Case ID	Name	Description
001	Start Page	Customer opens app and leads to login page
002	Login	Customer logs in and leads to users home page
003	Map	Customer's map displays all registered and active vendors near-by
004	Filter	Customer can filter near by vendors by categories
007	New User	Customer clicks create account and takes them to new user page
008	Social Media	Vendors can link their social media account on their bio.
009	Photos	Customer can post photos to vendor's page
011	Favorites	Customer's can add vendors to favorites list
012		
013	New Vendor	Vendor clicks New Vendor and is redirected to sign up page
014	Schedule	Vendor can set up business hours and appear active to customers
015	Live	Vendors can go live and appear on customer's

		maps or go offline and disappear on customer's maps	
018	Pins	Vendors show up with different icons depending if they have carts, stands, or trucks.	

4. Development Tools:

- Framework: React Native
- Languages: Javascript
- Code Editor: Visual Studio Code
- Backend Language: Fire Base
- GUI Builder: Code from scratch
- Server:
- Libraries: https://codingislove.com/top-15-react-native-libraries/

5. Estimated Scope:
*Non-Highlighted functions will be completed this semester

		Schedule		
Use Case ID	Name	Description	Sprint	
001	Start Page	Customer opens app and leads to login page	2	
002	Login	Customer logs in and leads to users home page	4	
003	Map	Customer's map displays all registered and active vendors near-by	6	
004	Filter	Customer can filter near by vendors by categories	7	
007	New User	Customer clicks create account and takes them to new user page	4	
008	Social Media	Customer can sign in with social media	3	
011	Favorites	Customer's can add vendors to favorites list	12	

013	New Vendor	Vendor clicks New Vendor and is redirected to sign up page	3
014	Schedule	Vendor can set up business hours and appear active to customers	5
015	Live	Vendors can go live and appear on customer's maps or go offline and disappear on customer's maps	5
018	Pins	Vendors show up with different icons depending if they have carts, stands, or trucks.	9

Constraint ID	Name	Description
CON0001	Language	Javascript
CON0002	Code Editor	VSCode
CON0003	Framework	React Native
CON0004	Database	SQL
CON0005	Libraries	https://codingislove.com/top-15-react-native-libraries

6. Estimated Size:

Total SLOC	Name	Source Lines Of Code (SLOC)	Produced SLOC
300	Start Page/Login	300	276
550	Home	250	160
1050	Map	500	143
1350	Filter	300	
2150	New User	200	164
2250	Social Media	100	

2350	Photos	100	
2550	Report Photos/ Reviews	200	
2750	Favorites	200	
3450	New Vendor	500	194
3950	Schedule	500	
4050	Live	100	58
6050	Pins	300	

Estimated size for the "REP APP" release v1.0.0 is roughly 3,800 lines of codes.

7. Identifications of Preliminary Risks:

	Risk	Impact	Decrease Probability	Status
001	Team member does not meet deadlines/quits (schedule change)	schedule delay, re-assign responsibilities, each team member would have more responsibilities	Maintain constant communication with regards to updates or changes. Address problems early.	N/A
002	similar app coming out with better features	App release may not be as successful/ fewer users.	Remain up-to-date on any applications similar to the one being designed, include better features.	N/A
003	IT Troubleshootin g	schedule delay. frustration within team,	Work as a team to identify a quickly find a	"completed"

	hardware/software	stress.	solution to hardware/software problems	
004	Team inexperienced with required tools	full features may not be taken advantage of. progress delay. implementing functionalities becomes more difficult.	Each team member should help/educate others on how to use the tools required for the project before any coding is done. Schedule could be modified to meet deadlines.	"in progress"
005	Developing wrong functionalities	loss of time. code must be re-written. Application release date gets delayed.	communicate with each team member to make sure that the correct functionalities are being implemented	N/A
006	Loss of Data	Most if not all Progress with project is lost. less time to complete project/failure to release project on schedule.	save all data in a cloud or in a portable storage device.	N/A
007	bugs during integration/ runtime (different phone versions)	take time to identify application compatibility with different phone versions. identify coding errors.schedule	finish tasks on time according to the schedule. Include extra within the schedule time to fix bugs	"in progress"

		delay		
008	difficulty fixing errors	team members focus on fixing errors rather than completed tasks and functionalities.	Understand code, explain what is going on to each team member.	"in progress"
009	changing requirements	More work is created, tight schedule , new requirements may be incompatible/inconsistent with current functionalities.	analyze and make sure that every requirement is necessary and makes sense.	N/A
010	not enough features	Must include more features, make sure features are compatible, design must be modified to fit new features.	include extra features only if project is complete or nearly finished.	N/A
011	incorrect design	Design may be inconsistent with functionalities. plan new design. Schedule delay.	design should be discussed with all members. Team should agree on a particular design. Design should meet user requirements and functionalities.	N/A
012	Not enough resources, Resources too expensive	team is unable to finish tasks.	resources to be used must be planned out before building the project.	"in progress"

			members should maintain realistic expectations for resources.	
013	Change of tools for design	Team must be re-educated to work with the new tools. Time is lost. Schedule delay.	The tools being used must be able to execute all of the functionalities that will be used within the program. Any new/extra tools being used must be compatible with current tools.	N/A

8. Description of the Project Asset Library (PAL)
- Using github for code sharing and version control

- Using google docs for sharing word documents among team members
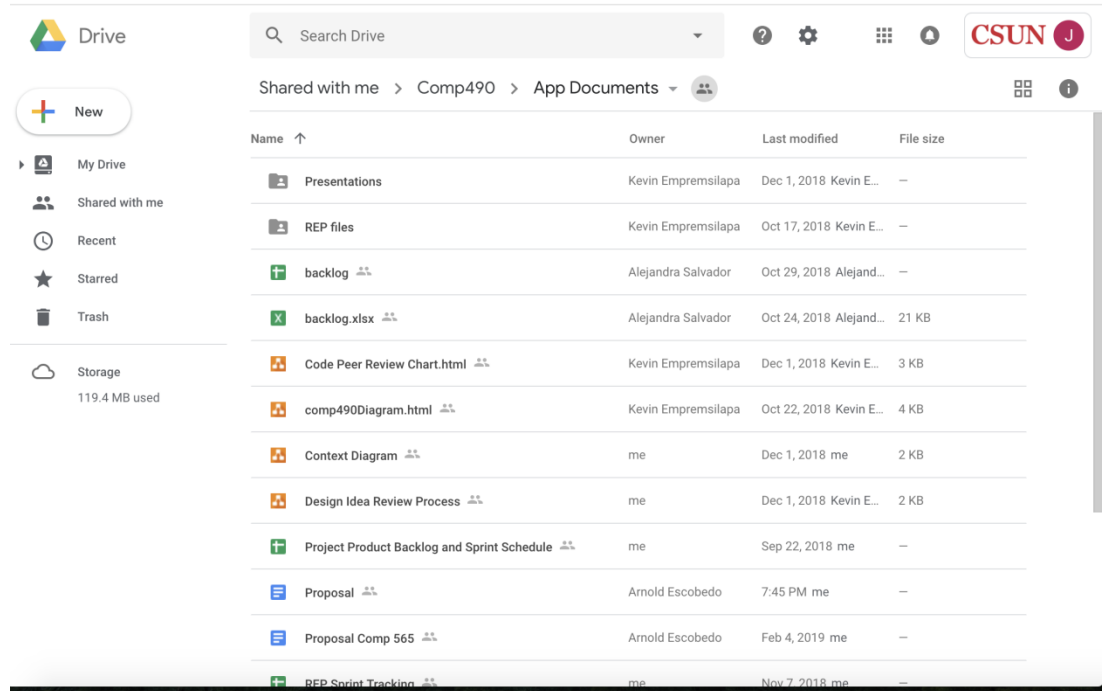
9. Description of how project progress will be tracked both at the project and sprint level i.e. Management tool (s) name (e.g. MIT Spreadsheet), description, and usage.

- Trello for documenting sprints and burndown charts
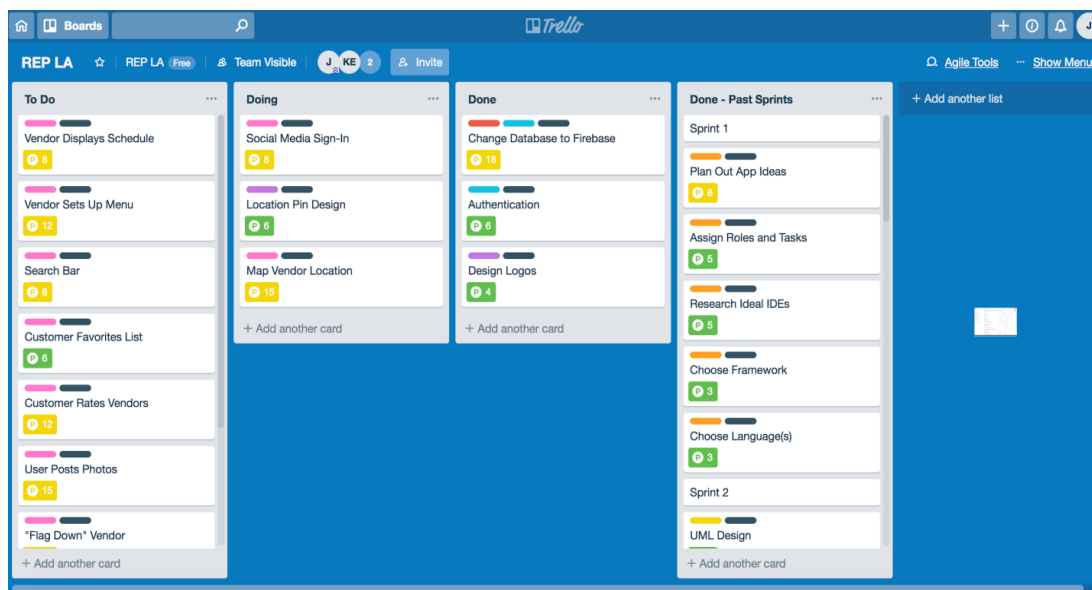
10. Project development process description--do not describe generic Scrum, but <u>specifically</u> how your team will execute this project e.g. team organization/roles, what/when planning will be done, description of managing the product backlog, releases, sprints, and retrospectives (reviews), artifacts/documentation prepared, etc. (This section can be brief for this first iteration – you will add more later.)

- team roles
  - FrontEnd
    - Team Member 1
    - Team Member 2
  - BackEnd
    - Arnold Escobedo
    - Team Member 3
- when planning will be done
  - planning will be done during the first two sprints

- description of managing product backlog, sprints and releases.
  - product backlog and sprints would be managed using Trello.
  - releases will be controlled over github.

- Peer Review
  - Deliverables are placed under 1 of 4 columns based on their status
    - Ideas, Under Review, Approved, Denied
  - Status of deliverables is determined during team meetings
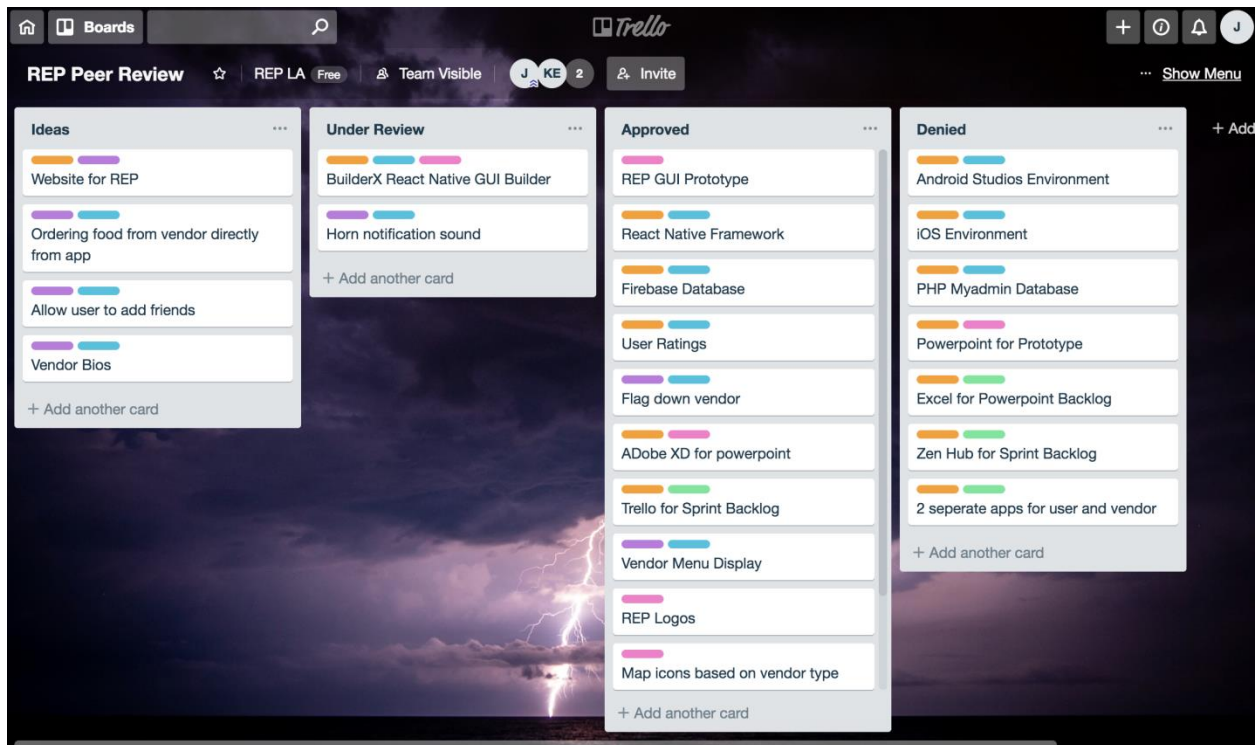
11. Changes made in Spring 2019
- No longer utilizing BuilderX for GUI due to financial and software glitch reasons
- GUI will be created from scratch using JavaScript, as well as the use of open-source libraries
- Functionalities removed due to time consumption creating the GUI without BuilderX
  - Vendor menu setup
  - "Flag" a vendor
  - Vendor reviews
  - Customers and Vendors posting photos
  - Food ratings
  - Credit card purchase and authentication