Sprint 1 Plan

Product Name: CruzCal
Team name: CruzCal Developers
Sprint Completion Date: Tuesday, April 19, 2022
Revision number: 1.3
Revision date: Thursday, April 19, 2022

Goal

The goal for sprint 1 is to establish a bare bones skeleton infrastructure for CruzCal. This includes setting up the architecture and processes for our application and workflow. For the first sprint, the goal is to have the team become familiar with the product idea, technologies, and workflow. Meaning there will be a significant amount of time spent on spikes, allowing team members to gain confidence and provide them with time for research with the hope that in the next sprint more focused work can be achieved.

User Stories for Release

1. As a student, I want to search for available courses during the term I'm interested in.
    a. **Tasks**
        i. Determine which CSS framework front end will use
        ii. Determine the testing front end uses
        iii. Determine the testing back end uses
        iv. Deploy web app using AWS or GCP
        v. Create information pane / card component:
            1. Subject
            2. Class Number
            3. Reset Button
            4. Add Button
            5. Info pop-up
        vi. Site Layout via flex
        vii. Pull-down for cards
2. As a student, I want to see course information for classes that I enter.
    a. **Tasks**
        i. Learn SlugSurvival API
        ii. Research PG ORM libraries
        iii. Decide request/response body structure for endpoints
        iv. Create API schema for validation
        v. Setup app template initializing server and registering routes
        vi. Endpoint to acquire class schedule for an array of classes (POST /schedule)

vii. Endpoint to get all valid terms (GET /terms)
viii. Endpoint testing
ix. Setup DB and create CourseInfo table with Sequelize
x. DB testing
xi. CRON job to build class data every day
xii. CRON job testing
xiii. Learn sequelize (spike)
xiv. Show validation errors under incorrect classes after submission

Sprint 1

User stories	Tasks	Name	Priority	Estimated Time	Start date	End date
As a student, I want to access CruzCal so that I can eventually use their services to manage my course schedule.	Determine which CSS framework front end will use	Andrew/Kitana	High	1	4/10/2022	4/11/2022
	Determine the testing front end uses	Andrew/Kitana	High		4/11/2022	4/19/2022
	Determine the testing back end uses	Pablo/Mario	High	0	4/12/2022	4/19/2022
	Determine how to split up with work for each developer	Everyone	Medium	1	on going	
	Create initial presentation	Everyone	high	2	4/10/2022	4/10/2022
	intialize front end enviroment	Kitana	high	4	4/9/2022	4/9/2022
	Deploy web app using AWS or GCP	Andrew	Low	1	4/15/2022	
	Setup Github Repo / file structure	Pablo			4/9/2022	4/9/2022
	Create information pane / card component: Term	Andrew	Medium	3	4/10/2022	
	Create information pane / card component: Subject	Kitana	Medium	3	4/10/2022	
	Create information pane / card component: Class Number	Andrew	Medium	3	4/10/2022	
	Create information pane / card component: Reset button	Kitana	Medium	3	4/10/2022	
	Create information pane / card component: Add button	Andrew	Medium	3	4/10/2022	
	Create information pane / card component: Info pop-up	Kitana	Medium	3	4/10/2022	
	Site Layout via flex	Andrew	Medium	3	4/10/2022	
	Pull-down for cards	Andrew/Kitana				
	Sprint plan and sprint release plan	Kitana	medium	3	4/10/2022	4/13/2022
As a student, I want to see course information for classes that I enter.	Learn SlugSurvival API	Backend	High	0	4/10/2022	4/10/2022
	Research PG ORM libraries	Backend	Low	0	4/11/2022	4/11/2022
	Decide request/response body structure for courses	Backend	High	1	4/10/2022	4/10/2022
	Create API schema for validation	Pablo	Low	2	4/11/2022	4/11/2022
	Setup app template initializing server and registering routes	Pablo	High	1	4/11/2022	4/11/2022
	Endpoint to acquire class schedule for an array of classes (POST /schedule)	Mario	High	5	4/11/2022	
	Endpoint to get all valid terms (GET /terms)	Mario	High	2	4/11/2022	
	Endpoint testing	Mario	Med	2		
	Setup DB and create CourseInfo table with Sequelize	Pablo	High	5		
	DB testing	Pablo	Med	2		
	CRON job to build class data every day	Tanmay	Med	3	4/11	4/15
	CRON job testing	Tanmay	Med	2	4/11	4/15
	Learn sequelize (spike)	Pablo	Med	1	4/12	
	Show validation errors under incorrect classes after submission		Med			

Fig.1: Task Board with priorities and time estimates

Team Roles

- Tanmay Mittal: product owner, backend developer
- Kitana Toft: SCRUM master, frontend developer
- Mario Reyes: backend developer
- Pablo Gaeta: backend developer
- Andrew Lim: frontend developer

Initial Task Assignment

Please reference figure 1.

1. Tanmay Mittal:
    a. Learn SlugSurvival API
    b. Research PG ORM libraries
    c. Decide request/response body structure for courses
    d. CRON job to build class data every day
    e. CRON job testing
    f. Determine how to split up with work for each developer
    g. Create initial presentation
2. Kitana Toft
    a. Determine which CSS framework front end will use
    b. Determine the testing front end uses
    c. Create information pane / card component: Subject
    d. Create information pane / card component: Term
    e. Create information pane / card component: Class Number
    f. Pull-down for cards
    g. Sprint plan and sprint release plan
    h. Determine how to split up with work for each developer
    i. Create initial presentation
3. Mario Reyes
    a. Learn SlugSurvival API
    b. Research PG ORM libraries
    c. Decide request/response body structure for courses
    d. Endpoint to acquire class schedule for an array of classes (POST /schedule)
    e. Endpoint to get all valid terms (GET /terms)
    f. Endpoint testing
    g. Determine the testing back end uses
    h. Determine how to split up with work for each developer
    i. Create initial presentation
4. Pablo Gaeta
    a. Learn SlugSurvival API
    b. Research PG ORM libraries
    c. Decide request/response body structure for courses
    d. Create API schema for validation
    e. Setup app template initializing server and registering routes
    f. Setup DB and create CourseInfo table with Sequelize
    g. DB testing
    h. Learn sequelize (spike)
    i. Determine the testing back end uses
    j. Determine how to split up with work for each developer
    k. Create initial presentation
5. Andrew Lim
    a. Determine which CSS framework front end will use
    b. Determine the testing front end uses
    c. Determine how to split up with work for each developer

d. Create initial presentation
e. Deploy web app using AWS or GCP
f. Create information pane / card component: Reset button
g. Create information pane / card component: Info pop-up
h. Create information pane / card component: Term
i. Create information pane / card component: Class Number
j. Create information pane / card component: Add button
k. Site Layout via flex
l. Pull-down for cards
m. Sprint plan and sprint release plan

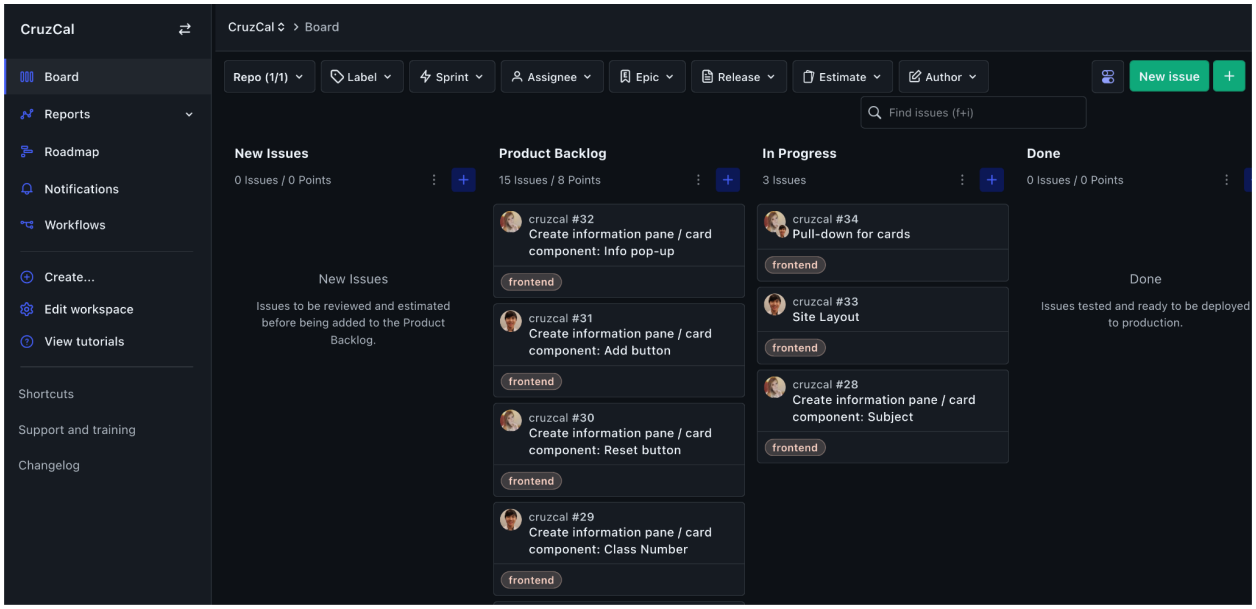
Initial Burndown Chart

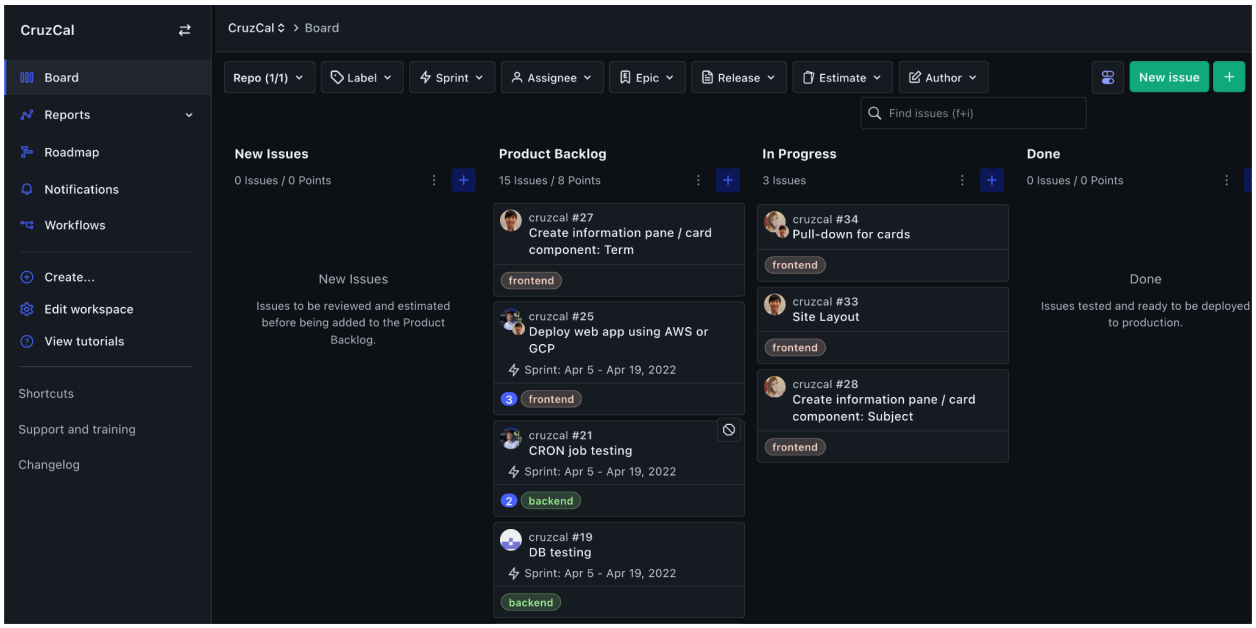
Our team decided to use a combination of *GitHub Issues* and *ZenHub* for the project management tooling for the life of this product. *GitHub Issues* is a project management productivity tool that can be used directly with GitHub to create project milestones, tasks, assign people and/or groups to tasks, and more. ZenHub is another productivity tool which links your team's GitHub project directly then automatically sets up task boards and generates reports. Below is one of the auto-generated burn down charts based on the issues for the current sprint.

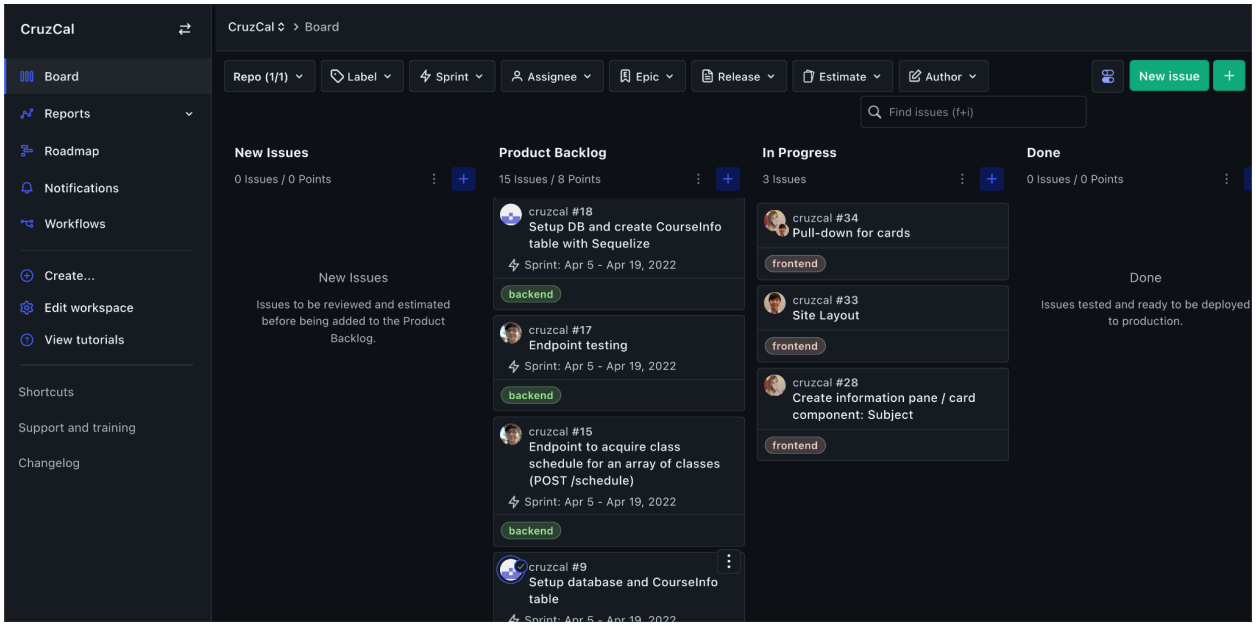
Fig.2: Link to Burndown Chart

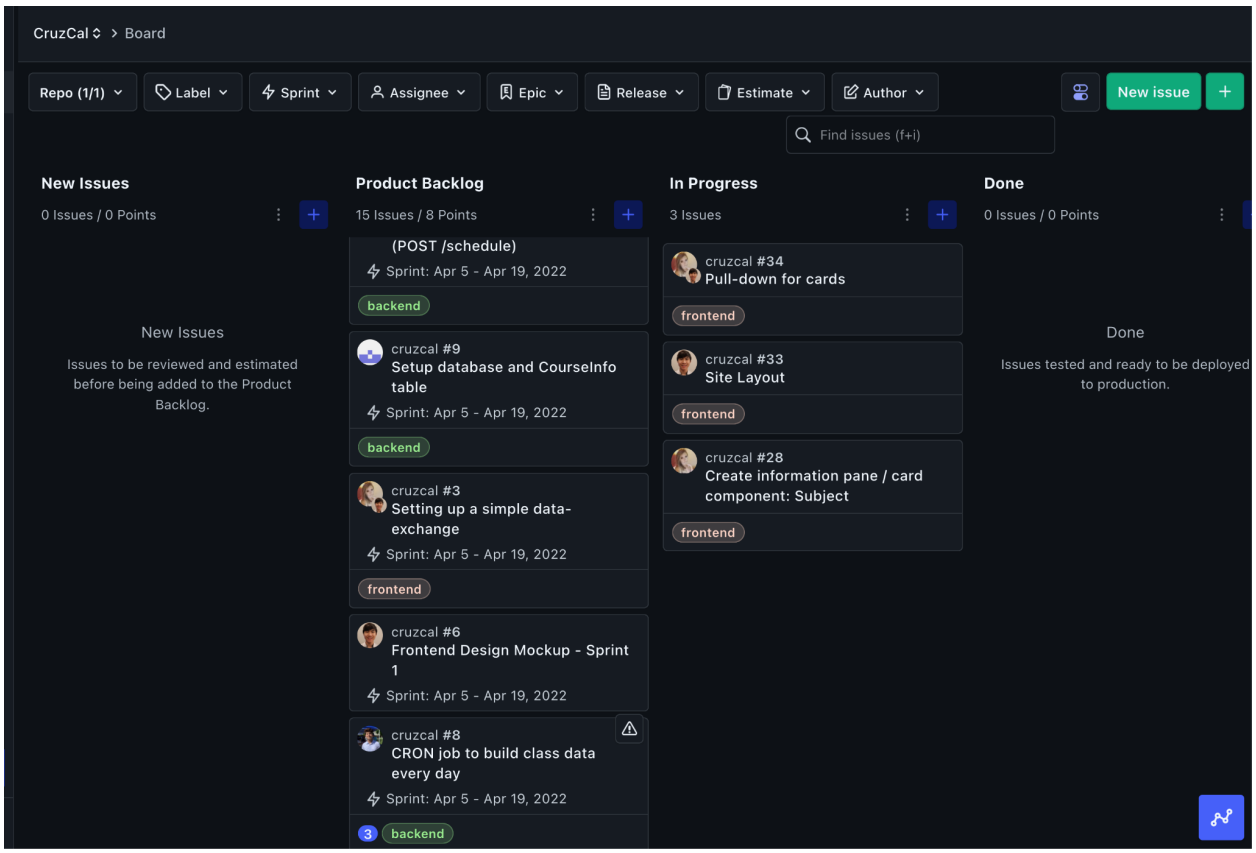
Initial SCRUM Board

The SCRUM board we have is created using both gitHub Issues and ZenHub. When we create an issue it is automatically linked to the ZenHub site, these issues can be categorized.









SCRUM Times

Weekly Meetings

- **Tuesday**: **10:30 - 11:30 am** meet on campus (or virtual) at LEEPs lab room 413
- **Thursday**: **5:30 - 6:15 pm** with TA (on campus or remote – TBD)

- **Friday**: **9:05 - 9:20am** in-person meeting 15 minutes after class (weekly)
- **Sunday**: **4:00pm - 4:15 pm** online check in for progress report (15 minute chat, virtual)

Check in Agenda

1. What progress has been made?
2. What complications are we facing?
3. How should we move forward with these facts?
4. What are the next steps to work on the current or following task?

How to Run Each Meeting

- 2 minute intervals for people to express & summarize their ideas or findings
- Developers should meet & determine how to split up the given tasks based on user stories per sprint