Housemates Sprint Reports

Sprint 1 Report

Housemates, Dev Team

Jack Luong, Daniel Carrera, Jackson Tran, Kyle Gong, Luciano Villacrucis
- 4/18/2022 -

Actions to stop doing:

- Not satisfied: Set back by some members getting sick

Actions to start doing:

- Plan out the meeting beforehand so we know what to talk about
- Have Scrum meetings more often

Actions to keep doing:

- Meeting weekly or whenever needed
- Communication over discord
- Keep using Jira to track issues and progress of sprints

Work completed/not completed:

Completed:

- **User story 1** ("As a *housemate, I want to be able to create and access my own profile to identify myself with other people*")
    1. Figma design login and signup (2 hours)
    2. Frontend: Create login page (2 hours)
    3. Frontend: Create signup page (2 hours)

- **Infrastructure Task 1** (As a *Housemates developer,* I want to *be able to use version control with my team and setup my coding environment,* so that *we can effectively develop code in parallel*)
    1. Setup Git and Github repository (2 hours)

- **Infrastructure Task 2** (As a *Housemates frontend-developer,* I want to *be able to setup my frontend environment in XCode and Swift,* so that *I can effectively develop frontend code*)
    1. Setup Swift and XCode for Frontend development (2 hours)
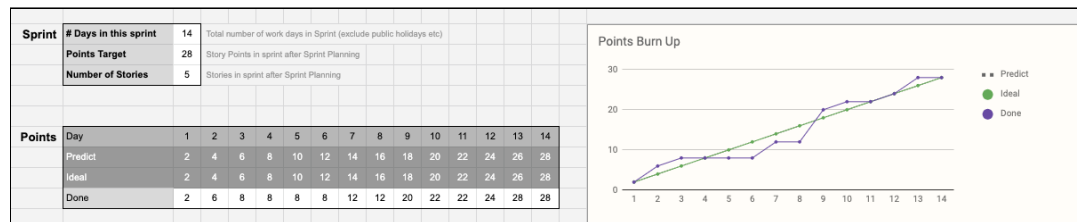   2. Sync initial frontend code in Github (2 hours)

- **Infrastructure Task 3** (As a *Housemates backend-developer,* I want to *be able to setup my backend environment in Python and Flask,* so that *I can effectively develop backend code*)
    1. Install tools: Python, Flask, IDE of choice (2 hours)
    2. Sync initial backend code in Github (2 hours)

- **Infrastructure Task 4** (As a *Housemates backend-developer,* I want to *be able to access a database and begin coding in parallel with my team,* so that *I can begin developing backend code using a database*)
    1. Choose a database hosting service (Google Cloud Platform) and configure it (4 hours)
    2. Setup backend API a skeleton infrastructure in Python Flask (4 hours)
    3. Connect database with backend skeleton (4 hours)

Not Completed:

-

Work Completion Rate:

- Total number of stories completed: 5
- Total number of work hours completed: 28 hours
- Total number of work hours not completed: 0 hours

- Total number of days during sprint:
    - 14 days
- Stories / Day:
    - 5/14 = 0.36
- Ideal work hours / day
    - 28 / 14 = 2
- Figures:

Sprint	# Days in this sprint	14	Total number of work days in Sprint (exclude public holidays etc)
	Points Target	28	Story Points in sprint after Sprint Planning
	Number of Stories	5	Stories in sprint after Sprint Planning

Points	Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	Predict	2	4	6	8	10	12	14	16	18	20	22	24	26	28
	Ideal	2	4	6	8	10	12	14	16	18	20	22	24	26	28
	Done	2	6	8	8	8	8	12	12	20	22	22	24	28	28

-

Sprint 2 Report

Housemates, Dev Team

Jack Luong, Daniel Carrera, Jackson Tran, Kyle Gong, Luciano Villacrucis
- 5/3/2022 -

Actions to stop doing:

- Not satisfied: Set back by some members getting sick

Actions to start doing:

- Keep backend API and database documentation up to date
- Assign team members to to-do APIs in API documentation

Actions to keep doing:

- Keep planning out meeting beforehand
- Meeting weekly or whenever needed
- Communication over discord
- Keep using Jira to track issues and progress of sprints

Work completed/not completed:

Completed:

- **User story 2** ("As a *housemate*, I want *to be able to signup and login to my Housemates application across accounts and sessions*, so that *I can securely and persistently access my housemates information*")
    1. Backend: create users table in database (2 hours)
    2. Backend: Implement *signup* API (2 hours)
    3. Backend: Implement *login* API (2 hours)

- **User story 3** ("As a *housemate*, I want *to be able to easily navigate through the application*, so that *I can easily access all the information in my application*")
    1. Figma design of *navigation control* (2 hours)
    2. Frontend: Create *customized segues and transitions* (2 hours)

- **User story 4** ("As a *housemate, I want to be able to view my profile and app settings so that I can see things such as my password, email, notifications…*")
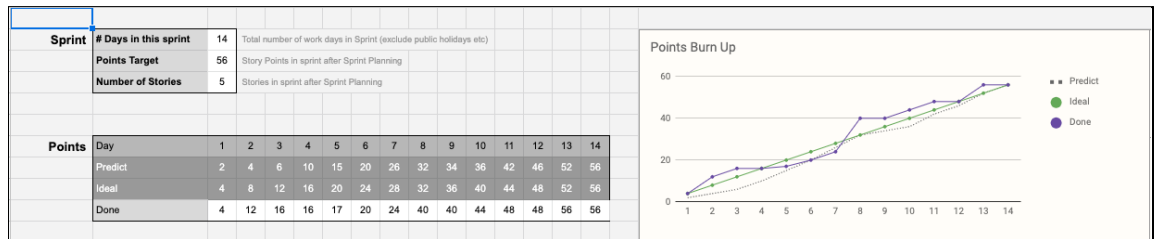   1. Figma design profile page (2 hours)

2. Frontend: Create profile page (2 hours)

- **User story 5** ("As a *housemate*, I want *to be able to view and add houserules to my house group*, so that *all housemates can have better understanding of the expectations around our living spaces*")
    1. Figma design of *view-house-rule* page (2 hours)
    2. Figma design of *add-house-rule* page (2 hours)
    3. Frontend: Create *view-house-rule* page (2 hours)
    4. Fronted: Create *add-house-rule* page (2 hours)
    5. Backend: Create house-rule table in database (2 hours)
    6. Backend: Implement *get-house-rules* API (2 hours)
    7. Backend: Implement *add-house-rule* API (2 hours)

- **User story 6** ("As a *housemate*, I want *to be able to view and add chores to my house group*, so that *all housemates can coordinate and work together around the house*")
    1. Figma design of *view-chores* page (2 hours)
    2. FIgma design of *add-chores* page (2 hours)
    3. Frontend: Create *view-chores* page (2 hours)
    4. Fronted: Create *add-chores* page (2 hours)
    5. Backend: Create chores table in database (2 hours)
    6. Backend: Implement *get-chores* API (2 hours)
    7. Backend: Implement *add-chores* API (2 hours)

- **User story 7** ("As a *housemate*, I want *to be able to view members in my house group*, so that *I can acknowledge and contact my housemates*")
    1. Figma design of *members-list side navigation* (2 hours)
    2. Frontend: Create *members-list side navigation* page (6 hours)
    3. Backend: Implement *get-user* API (2 hours)

- **Infrastructure Task 5** (As a *Housemates developer,* I want *to view, add, and edit API documentation, so that as a team we can keep track of all APIs, and understand their input and output*)
    1. Create users API table in google docs that specifies the following: type, path, params/data, status (2 hours)

- **Infrastructure Task 6** (As a *Housemates developer,* I want *to view, add, and edit with the database documentation,* so that *as a team we can keep track of the database schema*)
    1. Create database schema table in google docs that specifies the following: table-name, column-name, data-type, description (2 hours)

Not Completed:

-

Work Completion Rate:

- Total number of stories completed: 8
- Total number of work hours completed: 56 hours
- Total number of work hours not completed: 0 hours

- Total number of days during sprint:
    - 14 days
- Stories / Day:
    - 8/14 = 0.57
- Average stories per day:
    - (0.36 + 0.57) / 2 = 0.47
- Ideal work hours / day
    - 10 / 14 = 4
- Average Ideal work hours / day
    - (2 + 4) / 2 = 3
- Figures

Sprint	# Days in this sprint	14	Total number of work days in Sprint (exclude public holidays etc)
	Points Target	56	Story Points in sprint after Sprint Planning
	Number of Stories	5	Stories in sprint after Sprint Planning

Points	Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	Predict	2	4	6	10	15	20	26	32	34	36	42	46	52	56
	Ideal	4	8	12	16	20	24	28	32	36	40	44	48	52	56
	Done	4	12	16	16	17	20	24	40	40	44	48	48	56	56

Points Burn Up

- - Predict
● Ideal
● Done

-

Sprint 3 Report

Housemates, Dev Team

Jack Luong, Daniel Carrera, Jackson Tran, Kyle Gong, Luciano Villacrucis
- 5/17/2022 -

Actions to stop doing:

- Stop using form-data for POST & PUT requests

Actions to start doing:

- Keep Jira story progress up to date
- Assist on Frontend development

Actions to keep doing:

- Keep backend API and database documentation up to date
- Keep planning out meeting beforehand
- Meeting weekly or whenever needed
- Communication over discord
- Keep using Jira to track issues and progress of sprints

Work completed/not completed:

Completed:

- **User story 8** ("As a *housemate*, I want *to be able to assign, unassign, and edit a chore*, so that *I can edit chores and track what chores I must complete*")
    1. Frontend: Connect the backend chores APIs (4 hours)
    2. Backend: Implement *assign-chore and unassign-chore* API (4 hours)
    3. Backend: Implement *edit-chore* API (4 hours)

- **User story 10** ("As a *housemate*, I want *to persistently signup and login to my application*, so that *my information can be saved across multiple sessions*")
    1. Frontend: Connect the *signup & login* APIs (6 hours)

- **User story 13** ("As a *housemate*, I want *to persistently edit my profile*, so that I *can change my personal information across accounts and sessions*")
    1. Frontend: Connect the *user-profile* API (6 hours)
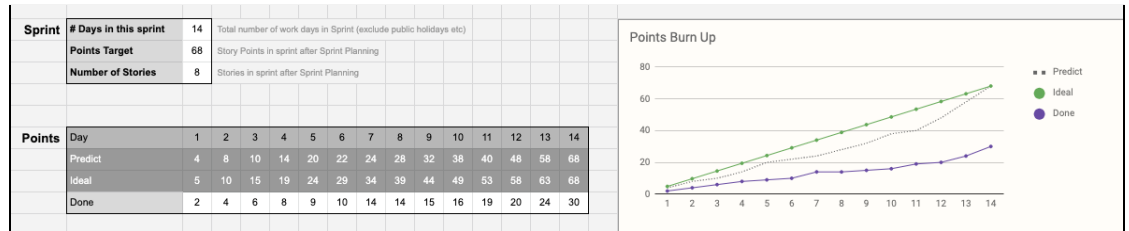   2. Backend: Implement *update-user* API (6 hours)

Not Completed:

- **User story 9** ("As a *housemate*, I want *to be able to delete chores*, so that *we can clear chores that have already been completed"*)
    1. Frontend: Connect the backend chores APIs (4 hours)
    2. Backend: Implement *delete-chore* API (2 hours)

- **User story 11** ("As a *housemate*, I want *to be able to edit house rules*, so that can change *house rules across many user sessions"*)
    1. Frontend: Connect the house-rules API (4 hours)
    2. Backend: Implement edit-house-rule API (4 hours)

- **User story 12** ("As a *housemate*, I want *to be able create, join, and leave a house group*, so that I manage a homegroup, be part of a house group, and leave a house group when I want*"*)
    1. Frontend: Connect the house group API (4 hours)
    2. Backend: Implement the *join & leave* house group API (2 hours)
    3. Backend: Implement the *create* house group API (2 hours)

- **User story 14** ("As a *housemate*, I want *to persistently view and interact with the home page*, so that I *can view the chores and house rules across accounts and sessions"*)
    1. Frontend: Connect the *chores and house-rules* API (4 hours)

- **User story 15** ("As a *housemate*, I want *to share and see my housemates' schedules*, so that I *can coordinate with my housemates based on their availability."*)
    1. Figma: Design the *schedule* page (2 hours)
    2. Frontend: Create the *schedule* page (6 hours)
    3. Backend: Implement the *schedule* API (4 hours)

Work Completion Rate:

- Total number of stories completed: 3
- Total number of work hours not completed: 38 hours
- Total number of work hours completed: 30 hours

- Total number of days during sprint:
    - 14 days
- Stories / Day:
    - 3/14 = 0.21
- Average stories per day:
    - (0.36 + 0.57 + 0.21) / 3 = 0.38

- Ideal work hours / day
    - 30 / 14 = 1.86
- Average Ideal work hours / day
    - (2 + 4 + 2.14) / 3 = 2.71
- Figures

Sprint	# Days in this sprint	14	Total number of work days in Sprint (exclude public holidays etc)
	Points Target	68	Story Points in sprint after Sprint Planning
	Number of Stories	8	Stories in sprint after Sprint Planning

Points	Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	Predict	4	8	10	14	20	22	24	28	32	38	40	48	58	68
	Ideal	5	10	15	19	24	29	34	39	44	49	53	58	63	68
	Done	2	4	6	8	9	10	14	14	15	16	19	20	24	30

Points Burn Up

- - - Predict
● Ideal
● Done

-

Sprint 4 Report

Housemates, Dev Team

Jack Luong, Daniel Carrera, Jackson Tran, Kyle Gong, Luciano Villacrucis
- 5/31/2022 -

Actions to stop doing:

- Missing SCRUM meetings
- Forgetting to make the new SCRUM master

Actions to start doing:

- Communicate more for front-end and back-end

Actions to keep doing:

- Keep Jira story progress up to date
- Assist on Frontend development
- Use JSON data for POST & PUT requests
- Keep backend API and database documentation up to date
- Keep planning out meeting beforehand
- Meeting weekly or whenever needed
- Communication over discord
- Keep using Jira to track issues and progress of sprints

Work completed/not completed:

Completed:

- **User story 9** ("As a *housemate*, I want *to be able to delete chores*, so that *we can clear chores that have already been completed"*)
    1. Frontend: Connect the backend chores APIs (4 hours)
    2. Backend: Implement *delete-chore* API (2 hours)

- **User story 12** ("As a *housemate*, I want *to be able create, join, and leave a house group*, so that I manage a homegroup, be part of a house group, and leave a house group when I want*")
    1. Frontend: Connect the house group API (4 hours)

2. Backend: Implement the *join & leave* house group API (2 hours)
3. Backend: Implement the *create* house group API (2 hours)

- **User story 14** ("As a *housemate*, I want *to persistently view and interact with the home page*, so that I *can view the chores and house rules across accounts and sessions*")
    1. Frontend: Connect the *chores and house-rules* API (4 hours)

- **User story 16** ("As a *housemate*, I want *to be able to vote for which house rule I want implemented* so that *I can have a say on the foundations that my housemates should follow*")
    1. Frontend: Connect voting APIs (4 hours)
    2. Backend: Implement the update_*house_rule*_voted_num API (6 hours)

- **User story 17** ("As a *housemate*, I want *to be able to see all the house rules I haven't voted on*, so that *I can have a chance to vote on every single house rule*")
    1. Frontend Connect voted rule APIs (4 hours)
    2. Backend: Implement the get_unvoted_*house_rules API* (6 hours)

- **User story 18** ("As a *housemate*, I want *to be able to see the house rules that have been approved and haven't been approved*, so that *I can see which house rules the majority of my housemates*").
    1. Frontend: Connect approval rule AP
    2. Is (2 hours)
    3. Backend: Implement the get_approved_house_rules API (4 hours)
    4. Backend: Implement the get_not_approved_house_rules API (2 hours)

- **Infrastructure Task 7** ("As a *housemate developer*, I want to be able to run unit tests on all my functions, so that I can determine the functionality of the frontend and backend systems")
    1. Frontend: Create unit tests for user actions (6 hours)
    2. Backend: Create unit tests for APIs (6 hours)

Not Completed:

- **User story 11** ("As a *housemate*, I want *to be able edit house rules*, so that *we can change house rules across many user sessions*")
    1. Frontend: Connect the house-rules API (4 hours)
    2. Backend: Implement edit-house-rule API (4 hours)

- **User story 15** ("As a *housemate*, I want *to share and see my housemates' schedules*, so that I *can coordinate with my housemates based on their availability.*")
    1. Figma: Design the *schedule* page (2 hours)
    2. Frontend: Create the *schedule* page (6 hours)

3. Backend: Implement the *schedule* API (4 hours)

- **User story 19** ("As a *housemate*, I want *to share and see my housemates' schedules across accounts and sessions*, so that I *can coordinate with my housemates based on their availability.*")
   1. Frontend: Connect *schedule* API (6 hours)
   2. Backend: Create schedule API (6 hours)

Work Completion Rate:

- Total number of stories completed: 7
- Total number of work hours not completed: 32 hours
- Total number of work hours completed: 58 hours

- Total number of days during sprint:
   - 14 days
- Stories / Day:
   - 7 / 14 = 0.5
- Average stories per day:
   - (0.36 + 0.57 + 0.21 + 0.5) / 4 = 0.41
- Ideal work hours per day:
   - 58/14 = 4.14
- Average ideal work hours / day
   - (2 + 4 + 2.14 + 4.14) / 4 = 3.07
- Figures

Sprint	# Days in this sprint	14	Total number of work days in Sprint (exclude public holidays etc)
	Points Target	90	Story Points in sprint after Sprint Planning
	Number of Stories	4	Stories in sprint after Sprint Planning

Points	Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	Predict	10	12	20	28	34	40	45	56	55	65	80	80	88	90
	Ideal	6	13	19	26	32	39	45	51	58	64	71	77	84	90
	Done	4	10	25	25	25	28	35	40	43	50	55	55	56	58

Points Burn Up

- -