

Team Noldea Sprint Reports

Bryan Kwong, Fuyu Zhang, Patrick Daniel Bustos

Introduction

Our team ran five 2-week long sprints to complete this project. We have sprint planning meetings on Sundays and start the sprint on the monday following. In that sprint meeting, we discuss what we hope to accomplish in the next two week and check our progress in the project.

We also have Sprint retrospectives to identify and resolve any inefficiencies we found in our sprint process. These sprint retros help us make changes and adapt the Agile framework that worked best for our project and our working style as a team. The following section discusses the changes that we made to best suit our need as a scrum team.

Modification to Agile framework

We acknowledge that these are not exactly according to requirement, but we feel that the goal of the agile framework is to evolve the process to one that works for the team. We felt that the base framework was a bit too strict for our needs and hindered our productivity. These changes allowed us to work more efficiently.

Daily stand ups

We initially met everyday on Zoom for our daily stand ups, but we found that most of the time, it was only a 2 minute meeting where we don't have much updates. We identified that our updates came in bursts of work due to work school and work schedule. Thus, we modified our stand up to be once every 3 days on Zoom, and once a day through messaging. This way we are more efficient with our time but can still reach out for help and provide progress daily progress updates

Task assignment

During our first sprint, we had used Jira to assign and manage our work. However, what we found out is that our working speed and efficiency varied widely throughout the project, thus there was no way to correctly predict the amount of work at the beginning of the sprint. Sometimes, we would finish early and not have anything to do, but other times we finish later and have to roll over our sprint. So for efficiency's sake, we stopped assigning work at the beginning and just agreed that we would take stories off the board as we work, this way, someone would always have something to do. This way, we operate more to complete the team's task, rather than just the individual's task. This system was inspired by the system currently being used at Bryan's workplace.

Component Leads

- Building frontend and then integrate with backend: Bryan
- AWS account manager and BookingService and HotelService: Nick
- LoyaltyService, UserService, HotelSearchService: Patrick

Sprint 1: Feb 28 - Mar 14

Sprint Planning

Sprint Goals:

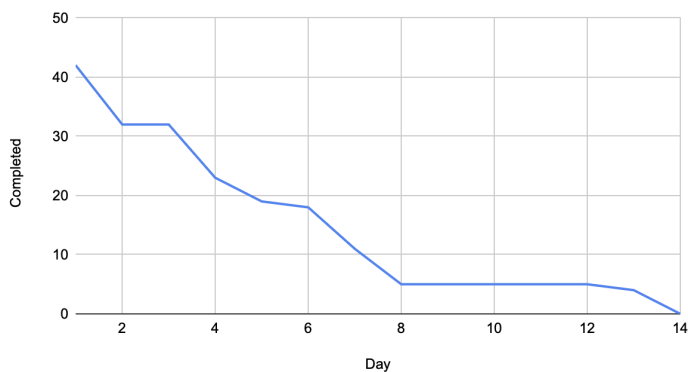
- Determine the tech stack that we will be using
- Complete designs and components
- Set up configuration for frontend and backend
- Create scaffolding for the frontend and backend

XP Core Value: Respect

- We respected each other's time by completing our tasks efficiently

Retrospective

Burn down chart



Ticket name	Estimates	Done
Create react app + dependencies	2	<input checked="" type="checkbox"/>
Connect react app to Amplify	2	<input checked="" type="checkbox"/>
Create AWS root and IAM account	3	<input checked="" type="checkbox"/>
BookingService: DynamoDB, API Gateway, Lambda	2	<input checked="" type="checkbox"/>
LoyaltyService: DynamoDB, API Gateway, Lambda	2	<input checked="" type="checkbox"/>
UserService: DynamoDB, API Gateway, Lambda	2	<input checked="" type="checkbox"/>
HotelService: DynamoDB, API Gateway, Lambda	2	<input checked="" type="checkbox"/>
BookingService: Diagrams and design	3	<input checked="" type="checkbox"/>
LoyaltyService: Diagrams and design	3	<input checked="" type="checkbox"/>
HotelService: Diagrams and design	3	<input checked="" type="checkbox"/>
BookingService: Diagrams and design	3	<input checked="" type="checkbox"/>
First time user experience with fields	3	<input checked="" type="checkbox"/>
Set up Cognito pool with Amplify	3	<input checked="" type="checkbox"/>
Set up routing and redirection on user type	3	<input checked="" type="checkbox"/>
Create base Hotel portal and connect	3	<input checked="" type="checkbox"/>
Create base Customer portal and connect	3	<input checked="" type="checkbox"/>
	42	

Stand Ups (once every 3 days)

*Despite only having official stands ups once every 3 days, we communicate continuously through messages

Mar 3

- Bryan: Create react app and downloaded all necessary dependencies. Work on connecting React apps to amplify next.
- Nick: Set up an AWS account with IAM users. Creating a base pipeline for Booking and Hotel service.
- Patrick: Create base pipeline for LoyaltyService, UserService, HotelService

Mar 6

- Bryan: Created routing and redirection. We need to prioritize API contract next sprint
- Nick: Create diagram and design for booking and hotel service. Shared it out during stand up, changes recommended from team

- Patrick: Create diagram and design for user, hotel, and loyalty service. Shared it out during stand up, changes recommended from team

Mar 9

- Bryan: Base route for hotel and customer portals. Code review and revisions
- Nick: First time user experience with fields. Code review and revisions
- Patrick: routing and redirection for user fields. Code review and revisions

Mar 12

- Bryan: Set up cognito user integration with app
- Nick: No updates
- Patrick: No updates

Sprint 2: Mar 14 - Mar 28

Sprint Planning

Sprint Goals:

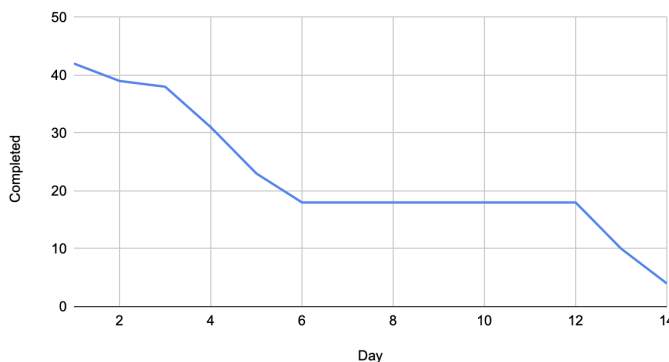
- Finish frontend architecture (routing, state, etc.) and common components
- Create and finalize API contract between frontend and backend
- Build UserService

XP Core Value: Simplicity

- Ensure that our REST API was simple to use and simple to build
- Ensure that the frontend architecture is scalable and utility common components

Retrospective

Burn down chart



Ticket name	Estimates	Done
Set up ReactRouter and Navigation bars	4	<input checked="" type="checkbox"/>
Set up Context React Slice for common info	4	<input checked="" type="checkbox"/>
Enforce user experience base on user type	5	<input checked="" type="checkbox"/>
Make call to user service	3	<input checked="" type="checkbox"/>
Set up API calling utilities	3	<input checked="" type="checkbox"/>
API Contract for BookingService	3	<input checked="" type="checkbox"/>
API Contract for UserService	3	<input checked="" type="checkbox"/>
API Contract for LoyaltyService	3	<input checked="" type="checkbox"/>
API Contract for HotelService	3	<input checked="" type="checkbox"/>
API Contract for HotelSearchService	3	<input checked="" type="checkbox"/>
UserService: Set up lambda	2	<input checked="" type="checkbox"/>
UserService: CRUD for Users	3	<input checked="" type="checkbox"/>
UserService: CRUD for User Credit Card	3	<input checked="" type="checkbox"/>
	42	

Stand Ups (once every 3 days)

*Despite only having official stands ups once every 3 days, we communicate continuously through messages

Mar 17

- Bryan: Set up react router and navigation bar, working on API contract for booking service
- Nick: Working on API contract for user service and hotel search service
- Patrick: Working on API contracts for hotel service and loyalty service. Start working on user service

Mar 20

- Bryan: Set up context react slice for common info. Integrate with user service and determine user experience base on user service response
- Nick: Finish API contracts and create API calling utilities. Critique and review API contract
- Patrick: Set up Lambda, CRUD for user and user hotel for user service

Mar 23

- Bryan: No updates

- Nick: No updates
- Patrick: Finish user service base on API contract

Mar 26

- Bryan: Make calls to the finished user service and check to make sure all routes work
- Nick: Make more API-calling utilities to simplify API calls
- Patrick: Update context slice to accommodate new user service format

Sprint 3: Mar 28 - Apr 11

Sprint Planning

Sprint Goals:

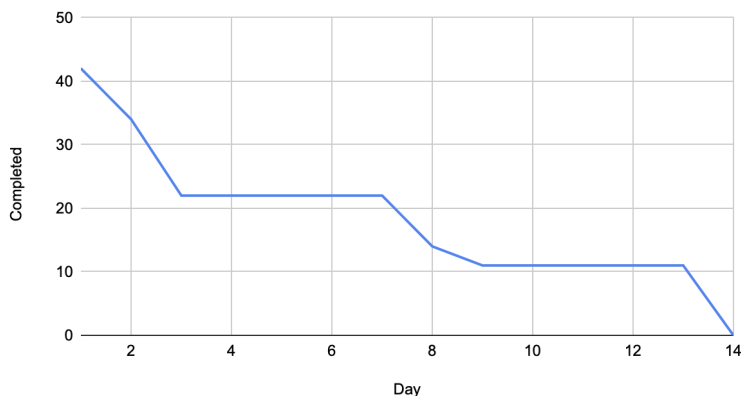
- Finish frontend scaffolding for Customer portal
- Build BookingService
- Build LoyaltyService

XP Core Value: Respect

- Respect each other enough to not micromanaged each other and have faith that we would execute our portion of the project according to the contract we set up

Retrospective

Burn down chart



Ticket name	Estimates	Done
Create customer base route	4	✓
Create booking scaffolding	4	✓
Create loyalty scaffolding	4	✓
Create existing reservation scaffolding	3	✓
BookingService: Base Lambda function configuration	3	✓
BookingService: CRUD for amenities	2	✓
BookingService: CRUD for rooms	2	✓
BookingService: CRUD for reservations	3	✓
BookingService: Create reservation route logic	2	✓
BookingService: Connect to LoyaltyService	2	✓
BookingService: Price the hotel reservation	2	✓
BookingService: Connect to UserService	2	✓
LoyaltyService: Base Lambda function configuration	1	✓
LoyaltyService: CRUD for loyalty account	2	✓
LoyaltyService: User service for Loyalty account	3	✓
LoyaltyService: Loyalty status management	3	✓
	42	

Stand Ups (once every 3 days)

*Despite only having official stands ups once every 3 days, we communicate continuously through messages

Mar 31

- Bryan: Create customer base routing
- Nick: Worked on Booking service base config and CRUD for amenities. Currently working on CRUD for rooms
- Patrick: Loyalty service configuration and CRUD. Will integrate with UserService

Apr 3

- Bryan: Bookings page scaffolding and functionality
- Nick: Finish CRUD for rooms. Working on CRUD for reservations

- Patrick: Loyalty status management and finished UserService integration, working on BookingService integration with LoyaltyService

Apr 6

- Bryan: Loyalty Scaffolding
- Nick: Finished and check functionality of all CRUD functionality for Loyalty and ReservationServices
- Patrick: Price hotel room management and reservation route logic

Apr 9

- Bryan: existing user scaffolding and integrate all different scaffolding into one route under /customer
- Nick: Integration test Loyalty, user and reservation service
- Patrick: Integration test Loyalty, user and reservation service

Sprint 4: Apr 11 - Apr 25

Sprint Planning

Sprint Goals:

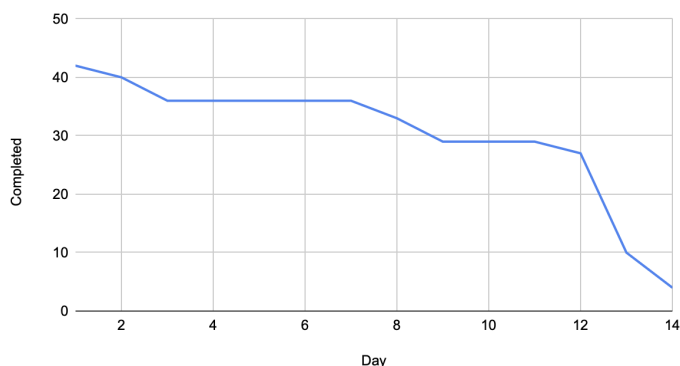
- Finish frontend scaffolding for Hotel portal
- Build RoomService
- Build HotelSearchService

XP Core Value: Respect

- Respect each other enough to not micromanaged each other and have faith that we would execute our portion of the project according to the contract we set up

Retrospective

Burn down chart



Ticket name	Estimates	Done
Create Hotel portal routing	5	✓
Create Reservation Scaffolding	4	✓
Create Branch scaffolding	4	✓
Create Room and Amenities scaffolding	4	✓
RoomService: Set up Lambda config	2	✓
RoomService: CRUD for amenities	3	✓
RoomService: CRUD for room	3	✓
RoomService: integrate with BookingService	3	✓
RoomService: integrate with HotelSearchService	3	✓
HotelSearchConfig: Set up Lambda config	2	✓
HotelSearchConfig: CRUD for new hotels	3	✓
HotelSearchConfig: Get all hotel search	3	✓
HotelSearchConfig: Integrate with BookingService	3	✓
	42	

Stand Ups (once every 3 days)

*Despite only having official stands ups once every 3 days, we communicate continuously through messages

Apr 14

- Bryan: Hotel portal routing and scaffolding for React redux
- Nick: Set up lambda for RoomService. Starting on CRUD for amenities and room
- Patrick: Set up lambda for HotelSearchConfigService. CRUD for hotel

Apr 17

- Bryan: Reservation scaffolding
- Nick: No updates
- Patrick: Get all hotels and integrate with BookingService. Test integration of existing routes and that it works with HotelSearchService

Apr 20

- Bryan: Branch management and room & amenities scaffolding
- Nick: Finish CRUD for room and amenities. Integrate RoomService with BookingService
- Patrick: Integrate RoomService with HotelSearchService

Apr 23

- Bryan: Integrate all scaffolding under /hotel route
- Nick: Finish integration with services
- Patrick: Start setting up some calling routes for integrating services into the frontend. Start on integration of user service.

Sprint 5: Apr 25 - May 9

Sprint Planning

Sprint Goals:

- Integrate frontend and backend
- Finalize documentation

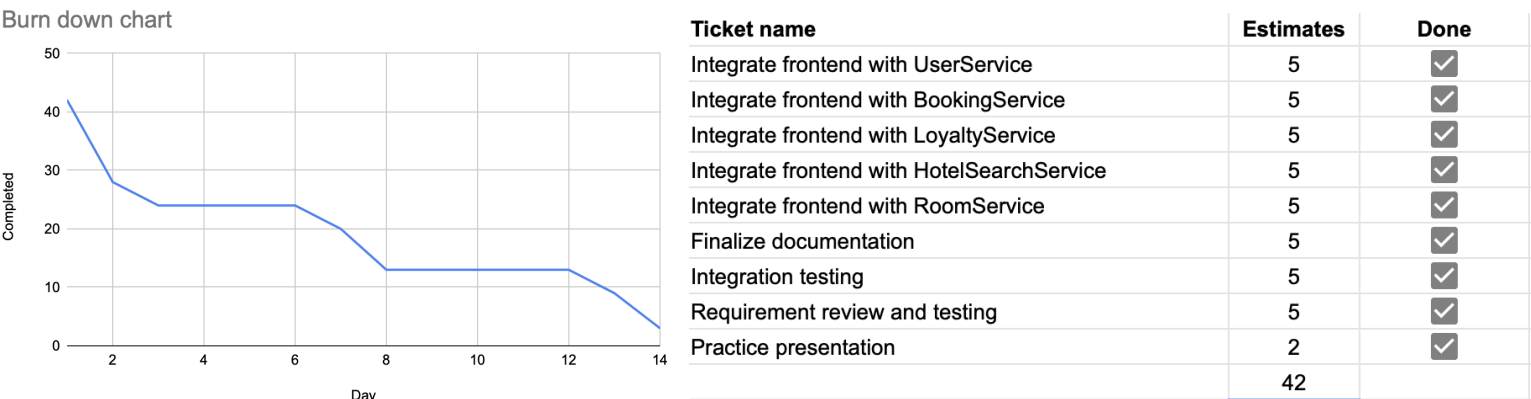
XP Core Value: Communicate

- Integration requires a lot of communicate between those who build the API and those consuming it in the frontend

XP Core Value: Feedback

- As we finish the project, we gather all the learnings from this project: what went well, what didn't and we will apply that to our next project
- Use AppSync or SpringBoot to streamline CRUD operation creations
- Use GatsbyJS to simplify frontend development

Retrospective



Stand Ups (once every 3 days)

*Despite only having official stands ups once every 3 days, we communicate continuously through messages

Apr 28

- Bryan: Drive integration of UserService and RoomService from frontend and backend
- Nick: Drive integration of UserService and RoomService from frontend and backend
- Patrick: Drive integration of UserService and RoomService from frontend and backend

May 1

- Bryan: Drive integration of LoyaltyService, HotelService, and RoomService from frontend and backend
- Nick: Drive integration of LoyaltyService, HotelService, and RoomService from frontend and backend

- Patrick: Drive integration of LoyaltyService, HotelService, and RoomService from frontend and backend

May 4

- Bryan: Drive integration of LoyaltyService, HotelService, and RoomService from frontend and backend
- Nick: Drive integration of LoyaltyService, HotelService, and RoomService from frontend and backend
- Patrick: Drive integration of LoyaltyService, HotelService, and RoomService from frontend and backend

May 7

- Bryan, Nick, Patrick: Review documents, integration testing, requirement check during integration testing, and practice presentation.