

Meeting 7

Oct 8th

- Priorities for this week & last minute work
 - [high] Get everything in dev
 - Code review
 - [low] Build sessions, permissions, integrate
 - With Amenities & Reports
 - [high] [sequence diagram](#)
 - Everyone does their services
 - Mondane-Monkeys compiles into big
 - Make teams post for each service
 - [high] architecture diagram
 - From a system POV & for each service
 - Same as Sequence (Mondane-Monkeys compiles)
 - [high] create a subset api document (for currently implemented features)
 - Mondane-Monkeys uses readmes to build that
 - Test on postman
 - [high] functionality only indicated for accounts server
 - Sprint 1 docs directy
 - [high] tests CI/CD pipeline
 - [medium] worksheet
 - Mention Cypress in worksheet
 - Mention minimal seams
 - Coverage reports
 - Leeroy for .net coverage reports
 - Most important tests
 - Add to worksheet document on teams
 - Acceptance + Integration : cypress
 - Reproduce team 7 environment
 - Rainclouded + Mondane-Monkeys

Meeting 6

Oct 1st, 2024

- TODO

- Sequence diagrams for all core features
- Once features are decided, tag related issues on repo
- Set up CI pipeline
 - Auto build
 - Code review
 - Regression test
- Testing plan
- Commitments
 - Rainclouded
 - Authentication
 - Tests
 - CI Pipeline
 - Nathan
 - Routing for login
 - React
 - angular
 - Simple login and home page front end
 - Leeroy
 - Finish incident reports
 - Sequence diagrams
 - Koala
 - Simple UI for Guest
 - Mondane-Monkeys
 - Review pull request

Meeting 5

Sept 28, 2024

- TODO
 - Update KANBAN –Leeroy
 - Update Worksheet - Koala
 - Create new Diagram - Mondane-Monkeys
 - Update main README - Rainclouded
 - Include project tree
 - Work on foundation implementation
 - Example frontend (amenities) – Nathan
 - Example (amenities) - Leeroy
 - Example DB (Tasks) - Mondane-Monkeys

- Priorities
 - Redesign core features + decide on focus for iteration 1
 - Enable stakeholder to decide focus features for sprint
 - Update Worksheet to reflect that
 - Diagram
 - Core features
 - Canaban
 - Update front facing readme
- Prototype implementation of everything
- Documents updates
 - Create readme with links
 - Including gradable material
 - Including directory tree + links to readmes
 - One line describing server objective
- Task heirarchy
 - Feature: [name] + label
 - includes checkboxes for US
 - US: amenities/[name] +Lebels
 - Includes checkbox for tasks
 - Dt: [name]
- Features
 - **Feature:** Task system (split?)
 - User side
 - Create tasks
 - **User Story:** As a hotel guest, I want to create service requests (e.g. room service, extra towels) so that my needs are promptly addressed
 - **Acceptance Criteria:**
 - **Scenario:** Guest creates a service request.
 - Given I'm a guest, when I open the guest task dashboard and choose to create a new service request, I can enter the request details (type of service, description, priority).
 - When I submit the request, it appears in my task list with a status of "pending," and the relevant staff members are notified.

- View task status
 - **User Story:** As a hotel guest, I want to view the status of my service requests so I can know when my needs will be met.
 - **Acceptance Criteria:**
 - **Scenario:** Guest views the status of service requests.
 - Given I'm a guest, when I open the guest task dashboard, I can see a list of my active requests with their status ("pending," "in progress," "completed").
- Staff side
 - View incomplete tasks
 - **User Story:** As a hotel staff member, I want to see a list of all incomplete guest requests so I can prioritize my work
 - **Acceptance Criteria:**
 - **Scenario:** Staff views a list of incomplete tasks.
 - Given I'm a staff member, when I open the staff task dashboard, I can see a list of all incomplete tasks with details such as request type, room number, and priority.
 - Claim a task
 - **User Story:** As a hotel staff member, I want to claim a task so that I can take responsibility for completing it
 - **Acceptance Criteria:**
 - **Scenario:** Staff claims a task.
 - Given I'm a staff member, when I open the staff task dashboard and click to claim a task, I am assigned as the responsible staff member, and the task status changes to "in progress."
 - Mark task complete
 - **User Story:** As a hotel staff member, I want to mark a task as complete once it's finished so that I can keep my work queue up to date
 - **Acceptance Criteria:**
 - **Scenario:** Staff Member completes a task

- Given I'm a staff member, when I am on the tasks staff dashboard, when I am viewing an existing task that I have just completed, I can click the "Completed" button and the system updates the task in real time and gives a confirmation alert.
- **Feature:** Accounts
 - Login (staff & guest)
 - **User Story:** As a user I need to be able to access the tools relevant to me so that I can make use of the system.
 - **Acceptance Criteria:** Users can log in with credentials and are directed to role-specific dashboards (guest, staff, manager).
 - Creation (staff accounts & room accounts)
 - **User Story:** As a hotel manager, I need to be able to give access to guests and employees so they can do their work
 - **Acceptance Criteria:** Hotel manager should be able to create and delete all staff and user accounts
 - **Scenario:** Creating a new account
 - Given I'm a logged in staff with permission to create staff accounts, when I fill in the necessary fields for user creation and submit it, a new account is created.
 - Permissions
 - **User Story:** As a manager, I need to be able to restrict access to various tools so that my services remain secure
 - **Acceptance Criteria:** Managers can assign or modify permissions for different user roles, controlling access to sensitive features.
 - **Scenario:** Editing permissions
 - Given I'm a logged in staff with permission to edit permissions, when I create a new group, configure the available permissions for that group, and assign accounts to that permission group, those accounts gain access to those permissions.
 - Settings
 - **User Story:** As a user. I want to be able to tailor the tools to my needs so that I can engage with the way that I want.

- **Acceptance Criteria:** Users can update settings (e.g. default dashboard views, notification preferences, and initial landing pages)
 - **Scenario:** Modifying preferences
 - Given I'm a logged in staff with permission to edit permissions, when I create a new group, configure the available permissions for that group, and assign accounts to that permission group, those accounts gain access to those permissions.
 -
- **Feature:** Amenities
 - View
 - **User Story:** As a guest, I want to access hotel amenity information so I can find out when amenities are available (eg, the hotel swimming pool is open)
 - **Acceptance Criteria:**
 - **Scenario:** Guest views hotel amenity information
 - Given I'm a guest, when I open the amenities guest dashboard, I can see a list of amenities offered by the hotel. Each amenity provides their operating hours, description and title.
 - Create, update, delete
 - **User Story:** As a staff member, I want to either create, delete or update hotel amenities so guests are always up to date with our amenity information.
 - **Acceptance Criteria:**
 - **Scenario:** Staff updates an amenity
 - Given I'm a staff member, when I open the staff amenities dashboard, and I am viewing an existing amenity, when I update the amenity details and click "Save", the system updates the report in real-time and shows a confirmation message.
- **Feature:** Incident reports
 - View
 - **User Story:** As a hotel manager, I want a way to view past and current incident reports created by my staff to better monitor the hotel incidents

- **Acceptance Criteria:**
 - **Scenario:** Hotel manager views past and current incident reports
 - Given I'm a hotel manager when I access the incident reports dashboard, then I can see a list of past and current incident reports created by my staff. Each report then includes details such as severity, status, and description.
 - Update
 - **User Story:** As a hotel manager, I want a way to update incident report details its relating case progresses in real time
 - **Acceptance Criteria:**
 - **Scenario:** Hotel Manager updates an incident report
 - Given I am a hotel manager, and I am viewing an existing incident report, when I update the incident details and click "Save", the system updates the report in real-time and shows a confirmation message.
- **Feature:** Business Analytics (TRIMMABLE)
 - No Guest
 - Staff
 - User Story: As a staff member, I want a way to view a guest's charges incurred in their stay so I could better review and verify their charges
 - Scenario: Given I'm a staff member, when I access the guest billing dashboard, I can select a specific guest's reservation info.
- **Feature:** Booking service (TRIMMABLE)
 - Guests
 - See available dates & rooms & prices
 - As a guest, I want a way to check room availability so I can plan my next hotel stay
 - As a guest, I want a way to view pricing details for hotel rooms so I could budget my stay
 - Staff
 - Create Room Rules
 - Manage reservation

Meeting 4

- Web application distribution server choice
- Development port convention?
 - Currently developer's discretion
 - Note, these ports are only for internal development and testing.
 - Staff: 4200
 - Guest: 5173
 - Accounts: 50001
 - Permissions: 50002
 - Sessions: 50003
- Comment coverage vs self-documenting
 - Examples:
 - Dockerfile

Meeting 3

- TA meeting
 - What to do still
 - When available
 - Thursday: 10:30
 - Friday: 12:30, 1:30
- Sprint goals
 - Create a “standards” doc within the repo (include naming conventions, branching strat, code review expectations)
 - Setting up azure environment
 - Mondane-Monkeys
 - Auth
 - Initially start with many permission requests
 - Permissions as a key will be implemented, but not necessarily used
- Task distribution & assignment\
 - Nathan + Koala
 - Build front end skeleton
 - Leeroy + Rainclouded
 - Build the services
 - Mondane-Monkeys
 - Build DB example
 - Build entire Tasks service

Meeting 2 Agenda

- Demo presentation (Timed)
- Public-ify repo
- Next steps
- CI/CD pipeline
 - Linting
 - Python – flake8
 - Js – ESLint
 - C# - sonarLint?
 - Webhook – teams
 - Automate Tests
 - Based on changed code section
- When to write code:
 - Mondane-Monkeys post demo into repo
 - Rainclouded will update server code
 - Nathan / Koala will mock up ui
 - Leeroy -> webhook
- Naming strats:
 - Git: branch_type/feature/issue_# (all lowercase)
 - feature/account_server/88989
 - Possible linters:
 - Js/ts prettier
 - Js/ts eslint
 - Python – flake8
 - Python – black
 - C# - sonarLint
 - Style guide:
 - Python <https://google.github.io/styleguide/pyguide.html>
 - Python <https://peps.python.org/pep-0008/>
 - C# - <https://google.github.io/styleguide/csharp-style.html>
 - Js/Ts <https://google.github.io/styleguide/jsguide.html>
 -

Meeting 1 Agenda

Sept 7, 10:00 AM

- finalize on name
 - Concierge
- Decide on tech stack (for presentation)
 -
- Core features
 - Login & auth
 - User Story: As a manager, I want to maintain profiles for my employees and guest so they can use appropriate services.
 -
 - Guest services
 - User Story: As a guest, I want I want to easily utilize hotel accommodation to improve my stay
 - Staff dashboard
 - User Story: As staff, I desire to view and respond to guest requests, so they have a better experience
 - User dashboard
 - User Story: As a user, I want to see information about my stay to help me stay organized
 - Non-functional
 - User Story: As a hotel owner, I want to support 500 users making a request over second for efficient access during peak times.
- Dev preferences
 - Leeroy
 - Backend
 - Rainclouded
 - DB
 - Github
 - Koala
 - Frontend
 - Nathan
 - Frontend
 - Mondane-Monkeys
 - Backend
 - DevOps

Meeting 1 Agenda

Sept 7, 10:00 AM

- Lock-In Project Idea
 - Rainclouded's Hotel description:
 - Wakeup call, amenity info
 - Based on room (not person)
 - Staff have their own access
 - Reservation
 - Staff directory
 - Reports
 - !Locked in!
- Contributions & Expectations
 - Regular Meeting
 - Petition 2 meetings
 - Standup on Tuesday
 - Distributing Work
 - Github Kanban
 - Sprint 0: jira & **prod manager** of each peice
 - Incomplete Work Protocol
 - Communication Expectations
 - Communication
 - Entirely through teams
- Version Control
 - Branching Strats
 - Code Review
 - Coding Standards
 - Task Board
 - Github kanban
- Architecture & Tools
 - Architecture:
 - Tentative Microservice:
 - TODO try getting the demo project
 - Tech Stack Options
 - Hosting Service
 - CICD Solutions
 - Documentation + Tests
 - Features
 - Spreadsheet to ideate
- Mondane-Monkeys's Microservice Demo

- Assign Team Leads

TODO:bv