Drone Case Study-My answers

Week2: Stakeholders

Use the system
- People receiving the parcel
- People who will load the drones
- Maintenance people
- People who monitor drones
- Cleaning staff
- Finance staff
- Retailer

Pay for the system
- Investors
- Company management(CEO)

Regulate the system
- CAA aviation regulation
- Local government(MCC)
- NATS-air traffic control
- Manchester airport
- Environment agency
- Information Commissioner's office
- Regulation for accessibility
- Prison Service

Affected By: interested in the system
- Drone manufacturer
- Local residents
  - -concerns about disruptions
  - -want to/want to use the service
- Landowners
  - -owner of warehouse
- Potential future investors
- Alternative delivery companies
  - -when we can't deliver due to weather
- Other air traffic
- Cloud computing provider

Week2- Requirements for the system

Eliciting requirements
- **-People receiving the parcels**
  Surveys, user interviews, feedback form
- **-People Loading the drone**
  Focus groups
- **-Maintenance staff**
  Workflow analysis, interviews
- **-Drone monitoring staff**
  Workshop and shadowing
- **-Cleaning staff**
  Informal interview and questionnaires
- **-Finance Staff**
  Audit and stakeholder meeting
- **-Investors**
  Presentation or structured interview
- **-CEO**
  Strategy session, high level interview
- **-CAA**
  Regulatory documentation, formal consultation
- **-MCC**
  Public consultations, regulatory meetings
- **-NATS**
  Coordination meeting and Technical workshops
- **-Landowners**
  Lease negotiation
- **-Manchester Airport**
  Airspace management consultations
- **-Information Commissioner Office**
  Data protection audit and compliance workshops
- **-Regulation and Accessibility**
  Accessibility audits, advisor board
- **-Prison service**
  Risk Assessment
- **-Drone Manufacturer**
  Feedback loop, Technical collaboration session
- **-Local Resident**
  Community meetings and surveys
- **-Investors**
  Investor pitches, QA sessions
- **-Alternative delivery companies**
  Partnership discussion
- **-Cloud computing provider**
  Technical requirements

Brainstorming requirements
People receiving the parcel
- -Delivery status tracking(Functional)
- -Set the location where they want to drop off
- -Reliable delivery time(Non-functional)

People loading the drones
- -Efficient loading interface(Functional)
- -Ergonomic loading process(Non-functional)

Maintenance Staff
- -Easy Access to diagnostics(Functional)
- -Spare parts Availability(Non-functional)

Drone monitoring staff
- -Real time drone tracking(Functional)
- -System uptime(Non-functional)

Finance Staff
- -Automated Financial reporting(Functional)
- -Financial standards(Non-functional)

Retailer
- -If the package has been delivered with estimated delivery time(Functional)
- -Alerts to fail delivery(Functional)

Fit Criteria
Delivery status Tracking
Test-Simulate the delivery and confirm the tracking real time using updates
Reliability Delivery Time
Metric-deliveries within the time frame
Package handling
Evident seals and audit during transportation
Real-Time delivery tracking
Live monitoring during test flight
Weather
Using an 3rd party API weather service to get the record of the wind speed and then make a conclusion

Prioritisation
Must Have
- -Data protection (Anonymity of the User)
- -Real-time drone tracking
- -Secure package handling

- -Compliance air traffic regulations
- -Proof of delivery image
- -Emergency drone recall feature

Should have
- Delivery status tracking
- Automated financial reporting

Could Have
- -Ergonomic loading process

- -Enhanced user interface for customers(In case of reporting)

Won't Have (for now)
- -Advanced AI-based delivery predictions

- -Preferred time of delivery

- -Drone personalization features

Week3- Design

Week 4 - Testing
Deadline -3 months
- **Retailer API Server(Medium)**
  Load Testing
  Unit Testing/Automatic Testing)
  Fuzzing
  Integration Testing
- **Package DB(High)**
  Load Testing
- **Tracking API Server(Critical)**
  Load Testing
  Integration testing
  Load Testing/Automatic Testing)
  Fuzzing
- **Mobile App(Medium)**
  User Acceptance Testing
  Manual Testing
- **Web App(Medium)**
  User Acceptance Testing
  Manual Testing
- **Scanner and Programmer(High)**
  User Acceptance Testing
  Unit Testing
  Integration Testing(Manual testing)
- **Internal DB(Critical)**
  Load Testing
  Integration Testing
  Load Testing
- **Control Server(Safety-critical)**
  Load Testing
  Integration Testing
  Load Testing
- **Monitoring System(Medium)**
  User Acceptance Testing
- **Maintenance System(Medium)**
  User Acceptance Testing
- **Financial Analytics(Low)**
  User Acceptance Testing
  Maintenance schedule
  When it will be?
  What actual maintenance will be done on the system?

Retailer API Server/Tracking API Server/Control Server/Package DB/Internal DB(Everyday)
- -Rotate log and delete old data
- -Apply security patches and software updates

Mobile App and Website(Bi-Weekly)
- -Fix UI bugs reported by users
- -Compatibility Testing
- -Optimize performance for better load times.

Drone Communication System(Monthly)
- -GPS Accuracy and response times
- -Land Immediately function works
- -Security protocols to prevent hacking attempts

Drone Hardware(Daily)
- -Inspect battery health and replace weak units.
- -Check propellers for damage.

Week 5-Implementation Methods
Cloud Based(Like GitHub, GitLab, Bitbucket)
Cloud based and self hosted
2FA, access control, audit logs
Easy setup and maintenance
Access from anywhere
Risk of third party security breaches
Costs increase with team use
Accessible via CLI,GUI, and web
Cost-$21/user/month (Enterprise pricing)
—————$126 Total

Self Hosted (Self Managed, SVN, Perforce)
Full control over security and compliance
No reliance on external services
Requires IT infrastructure and maintenance
Risk of internal server failure

SVN
- Self hosted or cloud
- Centralized, permission ,fine grained
- Free(self-hosted)
- Paid cloud plans
- Less flexible than Git

Mercurial
- Secure but less popular
- Cloud based
- Self hosted is free
- CLI based harder onboarding

Perforce
- Self hosted
- Enterprise grade security
- High cost enterprise planning- Need to get a quote
- Designed for large teams

Sajid and Backend API
The developer would be working on Mobile application and the Backend API for the overall system
- They can come up with a time which matches both UK and America
- Remote working using VS code Live share.
- They are senior developer and developer. They would interact as none require much mentorship

Sajid and Demitri(Web application)
- They both have front-end experience and can work together
- They would have similar work patterns
- Yes they can sit physically at same desk
- Different experience level and they can
Cara and Temi for database and Firmware installations

Deployment components
Backend API
Deployment Strategy
- Continuous Deployment pipeline with rolling updates to avoid downtime
- Version Consideration-API versioning to maintain compatibility with existing client.
- Rollback Plan: Blue green deployment or feature flagging to revert changes if necessary.
- API Documentation Update for developers
- Backend compatibility with existing clients
- Notify front-end about any changes.

Web Application
- Deployment Strategy: Progressive rollout using feature flags to test new features with a subset of users before full deployment
- Hosting-Cloud based deployment with CDN for performance optimization.
- User Training: Minimal as the UI is intuitive but a help section or guided tour.

Mobile Application
- Roll out on App store and google play store
- Ensure mobile app versions remain compatible with the back-end
- User Training in-app training on board
- Notification updates to user
- Any crash reports, feedback and quick fixes

Database
- Database migration with version control
- Backup before deployment and schema versioning for rollback if necessary

Firmware for Edge Devices
- Over the air(OTA) updates or manual updates for critical installations
- Pilot deployments before full rollout
- Provide fall time firmware version

Internal tools
- Ensure logs and audit trails, update to internal tools ensure minimal description
- User Training or workshops on the new features.

Week 4 -Project Management
Task1-What are task?
Retailer API Server
Task: API to communicate with the Package Database
Endpoints like
-POST/package
-Get /package(Id)
-PUT /package(Id)
-DELETE /package(Id)
-POST /track/update
Tracking API Server
Task-Enable real-time package tracking for mobile/web applications
-GET/track/{packageId}
-POST /track/update
Control Server
Task-Manage drone operations and interact with databases and external APIs.
-POST/drone/comm
-GET/drone/status{Id}
-GET/weather
-POST/maintenance/schedule

Weather Provider API(External Service)
Task-Provide real-time weather data
Endpoints
-GET/weather? Location=(lat,long)
Financial Analytics API
-POST /analytics/record
-GET /analytics/report
Database Management
Setup: Package DB and Internal DB
Subtasks:
-Define schema for package storage(ID,location,status,etc.).
-Store drone telemetry data
-Integrate financial records

Mobile&Web App development
Task- Develop UI/UX for real time tracking and control
-Features
-Packaging tracking interface (GET/track/{packageID})
-Drone status monitoring GET /drone/status/{id}

Drone and Scanner Integration
Task- Enable connection between drone and control server
-Implement scanner to read package details
-Send Scanned data to Tracking API Server
-Allow drone to receive and execute commands from the Control Server

Monitoring and Maintenance System
Task- Develop automated monitoring for drones
-GET/monitor/status
-POST/maintenance/report

Testing and Deployment
Task: Perform unit tests, integration test and deploy APIs and databases
-Test API responses and error handling
-Automate deployment and taking updates
-Monitor database performance

Task2-Critical Path

Week7-Project Management
Task1-Writing user stories and creating project backlog

People receiving the parcel
1. As a parcel recipient, I want to track my parcel so that I know where it is and when it will arrive.
2. As a parcel recipient, I want to specify the delivery location so that my package reaches the correct address.
3. As a parcel recipient, I want my delivery to be quick so that I receive my parcel as soon as possible.
4. As a parcel recipient, I want proof of delivery so that I can confirm my package was delivered successfully.

Operations Team (Lead, Monitor, and Maintain Drones)
1. As an operations team member, I want to view drone health data so that I know when a part needs replacing.
2. As an operations team member, I want to receive notifications with the location if a drone fails so that I can address issues quickly.
3. As an operations team member, I want to report a faulty drone so that it is not used until repaired.
4. As an operations team member, I want to schedule maintenance for drones so that they remain operational.
5. As an operations team member, I want to view the maintenance history of drones so that I can track repairs and servicing.
6. As an operations team member, I want to set the destination of a drone easily so that I can optimize deliveries.
7. As an operations team member, I want to see the location of all drones so that I can monitor operations effectively.
8. As an operations team member, I want to instruct a drone to return to base or land immediately so that I can handle emergencies.

Online Retailer
1. As an online retailer, I want to check if a parcel has been delivered and view proof of delivery so that I can confirm successful shipments.
2. As an online retailer, I want to send the necessary data for parcel delivery so that drones can execute deliveries accurately.
3. As an online retailer, I want to know whether drone delivery is available for a specific address so that I can offer the service to customers.
4. As an online retailer, I want parcels to be delivered without damage so that customers receive products in good condition.
5. As an online retailer, I want to know which items can be delivered by drone so that I can offer the service for eligible products.

Task 2
Product Owner (PO): Cara
Reasons:
- Has prior management experience, meaning she is comfortable making decisions and leading discussions.
- Likely to have good communication skills, which is crucial for engaging with stakeholders and defining user stories.
- Has a technical background, so she understands development constraints but can also focus on business needs.
- Returning after a career break means she might prefer a role that leverages her experience but isn't as coding-intensive.

Scrum Master (SM): Temi
Reasons:
- Strong communication skills and team-player mindset, which is crucial for facilitating discussions and ensuring smooth collaboration.
- As a recent graduate, might not have deep technical expertise yet, but this can actually be a plus, as the Scrum Master doesn't need to be the most senior developer.
- Enthusiasm for databases suggests analytical thinking, which can help track sprint progress and identify blockers.
- Good fit for a servant-leadership role, helping the team without enforcing authority

Sprint Length Recommendation: 2 Weeks
For this newly-formed team, a 2-week sprint would be the best option because:
- It allows for frequent feedback loops and quick course corrections.
- The team is new, so shorter sprints will help them adjust to Scrum quickly.
- It keeps sprint planning and retrospectives short, reducing wasted time.
- Since estimation can be tricky for a new team, shorter sprints reduce the risk of committing to too much work.

Task 3-Planning First Sprint
Sprint Breakdown
User Stories to include in Sprint 1
For Parcel Recipients:
1. User Story 1: "As a parcel recipient, I want to track my parcel so that I know where it is and when it will arrive."
   - Why it's important: This is a core feature for the customers, providing visibility on their parcel's location and estimated arrival time.
   - Estimated Time: 3 days (requires integration between the front-end and back-end systems).
   - Team:
     - Temi (Junior Developer): Front-end work for tracking UI
     - Alex (Senior Developer): Back-end development for integrating the drone's location system.
2. User Story 2: "As a parcel recipient, I want to specify the delivery location so that my package reaches the correct address."
   - Why it's important: Customers must be able to enter their preferred delivery location for accurate deliveries.
   - Estimated Time: 2-3 days.
   - Team:
     - Dimitri (Junior Developer): Front-end for the form to select the location.
     - Sajid (Developer): Back-end to validate the location and integrate it with drone routing.
3. User Story 4: "As a parcel recipient, I want to see proof of delivery so that I can confirm my package was delivered successfully."
   - Why it's important: Providing proof of delivery increases customer trust in the system.
   - Estimated Time: 2 days.
   - Team:
     - Sajid (Developer): Back-end for handling photo uploads and storage.
     - Temi (Junior Developer): Front-end to display proof-of-delivery photos.

For Operations Team:
1. User Story 7: "As an operations team member, I want to see the location of all drones so that I can monitor operations effectively."
   - Why it's important: Real-time drone monitoring is critical for the operations team to track drone statuses and intervene when necessary.
   - Estimated Time: 3 days.
   - Team:
     - Alex (Senior Developer): Backend integration for real-time drone location updates.
     - Cara (Developer): Front-end development for displaying drone locations on a map.

For Online Retailers:
1. User Story 3: "As an online retailer, I want to know whether drone delivery is available for a specific address so that I can offer the service to customers."
   - Why it's important: The online retailer needs this functionality to verify whether a customer's location is eligible for drone delivery before processing an order.
   - Estimated Time: 2-3 days.
   - Team:
     - Dimitri (Junior Developer): Front-end development for checking address eligibility.
     - Sajid (Developer): Back-end logic for validating addresses.

First sprint vs Milestone Plan
-First Sprint Plan: The focus here is on delivering a **set of minimal, but functional features** that provide visible progress, such as parcel tracking, delivery location management, proof of delivery, and basic drone monitoring.
-First Milestone Plan: The first milestone plan would likely have a broader focus, potentially covering things like **initial project setup, team formation, defining technical architecture, and setting the development environment.**

Task 4
Scrum with Kanban is likely a suitable framework, but some adjustments might be necessary depending on the team's familiarity with the framework, the complexity of the tasks, and the technical challenges ahead.

Task 3-Gantt chart deployment
Website-https://www.onlinegantt.com/#/gantt
To open the gantt chart

Online
Gantt...