UC Punch Project Review

1. Introduction

- **Purpose of the System**: The objective of the UC Punch program is to develop a new system to replace the active UC Flex system. This new system will have several improvements over its predecessor, ranging from consistent time tracking, clock-in/out tracking, improved schedule reliability, along with other improvements.
- **Key Objectives**: The primary goals of the UC Punch program were to reduce downtime, enhance the accuracy and reliability of time tracking, improve clock-in/out functionality, ensure consistent scheduling, and foster better communication between team members and management.

2. Roles and Responsibilities

- **Team Members**: List the key roles and names of individuals involved.
    - Project manager: Johanan Mendoza Aguazul
    - Researchers: Fakhruddin Shaik, Sid Kakarlapudi, Sean Brennan, VenkatYuvaRaaj
    - Content Developer: Becca Cordray
    - Quality Assurance: Alex Brooksbank
    - Content Creators: Luis Begic (content)
    - Programmer: Pratham Rajesh Biyani
- **Responsibilities**:
    - Project manager: To effectively lead the project, delegating tasks to the appropriate individual(s), making sure that all tasks are completed accurately and on time.
    - Researchers: To gather and analyze relevant data, providing critical insights and recommendations to inform project decisions.
    - Developers: To design, build, and implement technical solutions that meet project requirements and deadlines.

- Content Creators: To develop engaging and impactful content that aligns with the project's goals and target audience.
- Programmer: To write, test, and maintain code to support the technical functionality of the project.

3. Project Budget:

Direct costs:

- Software Development - $100,000.
- Infrastructure Costs - $100,000.
- Licensing Fees - $50,000.

Indirect costs:

- Research and Analysis - $30,000
- Team training - $25,000
- Change Management Support - $25,000
- Communication tools - $20,000

Contingency Costs:

- Contingency Reserve - $175,000

4. Work Processes and Procedures

- **Task Breakdown**: Outline the steps involved in the work or project.

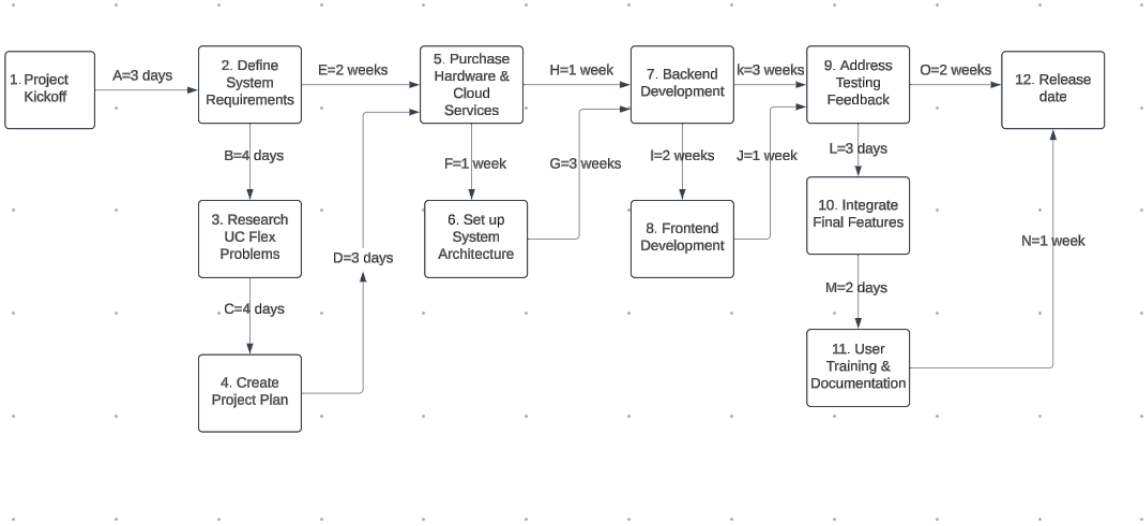
1. Planning and Preparation:
    - Brainstorming: Market Research
    - Requirements Gathering: Defining App Features
    - Budgeting and Funding: Competitor Analysis & Risk Assessment

2. Development:
    - Front-End Development: UI Design
    - Back-End Development: Create Wireframes, Implement UX Design, Set Up Database, Integrate APIs, Build Server-Side Logic
3. Testing, Debugging and Marketing:
    - Testing: Unit Testing
    - Quality Assurance: User Acceptance Testing
    - Pre-Launch Marketing: Performance Testing, Create Landing Page, Social Media Campaign, Reaching Out to Influencers
4. Launch:
    - Launch: Deploy App
    - Post Launch Support: Collect User Feedback, Bug Fixes, `
    - Enhancing New Features

- **Standard Operating Procedures (SOPs)**:
    - Example: SOP for Machine Operation, SOP for Safety Inspections, etc.

      Pratham Rajesh Biyani implemented tasks using Flask for web frameworks, Python for backend logic, and HTML for front-end structure, ensuring seamless integration. GitHub will manage source control, track updates, and facilitate collaborative teamwork throughout the development process.

- **Workflow Diagram**:



5. Resources and Equipment

- **Material Requirements**: Materials included access to Flask, HTML, Python, and GitHub platforms, as well as documentation or reference guides for these tools.
- **Equipment and Tools**: Tools used included laptops or desktops with sufficient processing power, reliable internet access, and Lucidchart for visual workflow creation and project mapping.
- **Resource Allocation**: Resources were distributed based on individual roles, with developers and programmers receiving the required software tools and licenses, while researchers and content creators had access to data analysis and content editing platforms.

6. Safety and Risk Management

Data Integrity Issues

- **Description:** During migration, data may become corrupt or incomplete, requiring additional validation efforts.
- **Potential Impact:** Extra costs for data validation, cleaning, and re-migration. Estimated impact: 10-20% of the data transfer budget. Could also impact the timeline causing setbacks.

Additional Security Requirements

- **Description:** New security standards or regulations may require more advanced encryption or security protocols, increasing costs.
- **Potential Impact:** Increased cloud service costs for enhanced security features (e.g., encryption, multi-factor authentication). Estimated impact: 5-10% of cloud storage costs.

Downtime During Migration

- **Description:** Unplanned downtime during data migration may delay access to critical systems and incur additional labor costs for troubleshooting.
- **Potential Impact:** Extended downtime could result in an additional 5-7% in labor costs and potential business interruptions.

Example Scenario:

Scenario 1: Additional Security Requirements

- **Situation:** New university-wide security protocols mandate the use of advanced encryption for all employee data.
- **Example:** Cloud storage vendors charge extra for enabling end-to-end encryption and integrating multi-factor authentication (MFA) for administrator access.
- **Mitigation Strategy:**
  - Evaluate cloud vendors' pricing structures beforehand and select one with scalable security options.
  - Conduct **security audits** to ensure compliance with regulations without unnecessary expenses.

Scenario 2: Downtime During Migration

- **Situation:** Unexpected system incompatibility causes delays, preventing employees from clocking in or out for an entire workday.
- **Example:** Employees are unable to record attendance, requiring manual logging by HR, resulting in errors and productivity loss.
- **Mitigation Strategy:**
  - Schedule migration during **off-peak hours** or weekends to minimize operational impact.
  - Establish a **fallback system** (e.g., manual timesheets) to ensure continuity during unplanned downtime.

Scenario 3: Physical Security of Servers

- **Situation:** On-premises servers hosting UC Punch are exposed to physical threats such as theft, fire, or unauthorized access.
- **Example:** A fire in the server room damages hardware, causing system outages and potential data loss.
- **Mitigation Strategy:**

- Install fire suppression systems, biometric access controls, and surveillance cameras in server rooms.
- Maintain regular **backups** of critical data in secure, off-site cloud locations.

Scenario 4: Insider Threats

- **Situation:** A disgruntled employee attempts to tamper with system settings or delete critical data.
- **Example:** A former developer with unauthorized access to GitHub repositories alters the application code.
- **Mitigation Strategy:**
  - Enforce strict **access controls** and revoke permissions immediately upon employee exit.
  - Use **audit logs** to monitor access and activity within the system.

Scenario 5: Compliance Violations

- **Situation:** The UC Punch system inadvertently stores personally identifiable information (PII) in non-compliant ways, violating data protection laws.
- **Example:** Timekeeping records include sensitive details (e.g., social security numbers) stored without encryption.
- **Mitigation Strategy:**
  - Ensure all data handling complies with **GDPR** or **HIPAA** standards as applicable.
  - Conduct **periodic compliance reviews** with legal and IT teams.

7. Communication Procedures

- **Internal Communication**: Define how communication will flow between team members.
  - All internal communication was done between each group member, then details were relayed to the PM by one or all parties involved. This communication was done both in person and through teams.
- **External Communication**: Define how and when communication will occur with external stakeholders.
  - Example: Client updates, progress reports, meetings.
- **Reporting Channels**: Describe how progress will be monitored and reported to supervisors and managers.

- Progress was monitored and reported largely in person. Some communication of major events or milestones communicated over teams and then discussed during the next in person meeting.

8. Time Management and Scheduling

- **Kickoff: (2024-09-24)**
- **Brainstorming: (2024-09-25 – 2024-10-08)**
- **Development: (2024-10-09 – 2024-10-29)**
- **Testing and Quality Assurance: (2024-10-30 – 2024-11-12)**
- **Finalizing: (2024-11-13 – 2024-11-26)**
- **Release: (2024-11-27)**

9. Quality Control and Monitoring

- **Quality Standards**: All work must adhere to industry best practices, company-specific guidelines, and any relevant compliance standards, such as ISO 9001 for quality management.
- **Inspection and Audits**: Regular internal inspections will be conducted by the Project Manager, with periodic audits by third-party professionals to ensure quality and compliance with set standards.
- **Feedback Loop**: A structured feedback system will be implemented where team members and stakeholders can submit feedback through designated channels, with follow-up actions taken by the project manager to address and resolve concerns promptly.

10. Documentation and Record-Keeping

- **Document Management**: Documents were created as needed and managed through Microsoft Teams, where they were stored securely and made accessible to authorized team members.
- **Reporting Formats**: Reports and records were standardized using a consistent format, including headings, sections, and clear data presentation for uniformity across all documentation.

- **Record Retention**: Documents and records were retained for a minimum of one-year post-project completion, or longer if specified by company or regulatory guidelines.

11. Performance Evaluation and Review

- **Monitoring Performance**: Performance was evaluated through milestone completion rates, timelines adherence, and task accuracy, with all progress consistently on schedule. The project manager periodically reviewed milestone completion and task accuracy to ensure timelines were met and progress remained on schedule.
- **Review Schedule**: Periodic reviews were conducted at each project milestone to assess progress, ensure alignment with goals, and identify areas for improvement.

12. Team Building Activities:

- Icebreakers and Introductions
- Problem Solving Activities
- Communication Games
- Role Swap Scenarios
- Team Building Workshops\
- Outdoor or Physical Activities
- Virtual Team Building Activities
- Reflection and Feedback Sessions

13. Conclusion

- **Summary of Project**: The goal of the UC Punch project is to develop a new timecard tracking system with improvements to the punch-in and punch-out accuracy, and revamp of the time reporting system. This would allow for the increased reliability of employee time tracking and pay accuracy, reducing the overall potential liability and revenue loss for companies using a similar system. We completed this by developing a web-based portfolio compiling all of the system goals and overall timeline a singular place alongside the development of the code.

- **Next Steps**: The UC Punch system will transition into the beta phase, during which user feedback will be gathered to identify areas for improvement and ensure functionality meets expectations. Based on feedback, adjustments and refinements will be implemented before final deployment.
- **Contact Information**:
    - Project manager: Johanan Mendoza Aguazul
    - Researchers: Fakhruddin Shaik, Sid Kakarlapudi, Sean Brennan, VenkatYuvaRaaj
    - Content Developer: Becca Cordray
    - Quality Assurance: Alex Brooksbank
    - Content Creators: Luis Begic (content)
    - Programmer: Pratham Rajesh Biyani