**BUSINESS CASE & PROJECT CHARTER**

**Project Name**

ProcrastiNOT

**Problem Statement**

In today’s world, staying productive often means staying organized. Whether it’s a movie watchlist, recommended books, or restaurants to explore on a date, keeping track of everything can be a challenge. ProcrastiNOT aims to simplify this by providing a sleek platform to manage your lists easily. Collaborate with friends and family on shared lists or copy their recommendations into your own collection.

**Market Research**

Most users simply have their lists saved in the default "Notes" app whether its on Android or iOS. This is a disadvantage, since these important lists can be easily lost amongst other notes users typically write in the "Notes" app. In addition, collaborating with other users to build a list is currently impossible, and lists may only be shared by fully copying them into a chat or something similar which is clearly a hassle.

There currently exists an app in the app store called "Listy" which executes our idea in a similar way, but limits you to items that exist in their database of books, movies, wines, food, tv shows etc. Clearly, this is an issue for users and we'd rather give them freedom and full functionality to manage any list they'd like.

**Purpose and Objectives**

The project is being undertaken to address the challenges of managing and organizing multiple personal and collaborative lists effectively. Modern users struggle to track recommendations and to-dos across various categories such as movies, books, restaurants, or travel destinations. ProcrastiNOT aims to centralize these lists into a single intuitive platform, encouraging users to stay organized, share recommendations, and collaborate. The primary objectives are:

* Provide a user-friendly app for managing unique lists.
* Enable collaboration and sharing features for shared-lists.
* Enable decision-making through engaging tools like random selection, voting, and progress tracking.

**Benefits and Impact**

* Users can manage multiple lists in one location, reducing clutter and confusion.
* Shared list functionality fosters better interactions between friends and family.
* Interactive features like “I Feel Lucky” and progress tracking make list management enjoyable.
* Intuitive UI and smart suggestions save users time and effort.
* Encourage productivity and better decision-making.
* Strengthen social connections through shared experiences.

**Scope**

Included:

* A mobile application for creating, managing, and sharing lists.
* Integration of features like random selection, notifications, and progress tracking.
* Support for categories such as entertainment, dining, and tasks.
* Enable user collaboration through QR Codes and Links that would be used to share lists.

Not Included:

* Contextual suggestions (e.g., GPS for restaurant lists) not supported in the initial release.
* The initial release focuses on personal and shared lists but excludes advanced integrations with external platforms (e.g., streaming services or delivery apps) or any feature that would require scraping the net.
* Only mobile platforms (iOS and Android) are supported in the initial release.

**High-Level Risk**

* **Problem:** Exposing user's security and privacy  
  *Solution*: Implement end-to-end encryption and strong privacy policies.
* **Problem:** Adding too many features may overwhelm users.  
  *Solution*: Conduct user testing and feedback and prioritize features important to said users.
* **Problem:** User retention  
  *Solution*: Engage users through gamification and provide regular updates based on user feedback.

**Alternative Solutions**

1. Standalone List Manager: Focus solely on personal lists without collaboration features.
2. Platform-Specific Solutions: Build an app tied to specific services (e.g., a watchlist for streaming, readlist for books etc).

Why ProcrastiNOT is required:

* A standalone list manager lacks the social and engaging aspects that make lists actionable.
* Platform-specific solutions limit versatility and miss the broader opportunity of centralized organization.
* ProcrastiNOT provides a flexible approach that meets the needs of both personal and the untapped social list management.

**Key Stakeholders**

**End-Users:**

Individuals managing personal and shared lists (primary focus).

**Team Members:**

Yoni Baruch, Koren Abdush, Yevgeny Ivanov

**Resources Required**

Team Expertise:

* Frontend and backend developers.
* Mobile app developers.
* UI/UX designers.
* Data Security Experts.

Technology:

* Development tools and frameworks .
* Cloud services for database and notifications (e.g. AWS).

Facilities:

* Testing devices for multiple platforms.
* Collaboration tools for remote team management.

**STATEMENT OF WORK**

**Vision Statement**

The vision of the ProcrastiNOT project is to provide a centralized platform for managing all personal and shared lists, making it easy for users to organize their tasks, recommendations, and experiences. The app should allow users to collaborate on lists, discover new things, and stay organized in a fun and engaging way.

**Scope of the Project**

* Development of a mobile app for managing and sharing lists across multiple categories (e.g., entertainment, food, travel).
* Features such as list creation, task tracking, collaboration, random selection tools, and notifications.
* Sensor integrations for contextual suggestions based on GPS and camera (e.g., suggesting nearby restaurants, scanning items).
* Cross-platform support for both iOS and Android.

**Scope of Work**

Design Phase:

* Develop visual prototypes for key features (list creation, sharing, and collaboration) using Figma.
* Define UI/UX to plan development.

Development Phase:

* Set up the mobile app architecture and back-end infrastructure (databases, APIs).
* Implement authentication features.
* Implement key app basic features (list creation, random selection, progress tracking).
* Implement key app shared-list features (collaboration through links/QR Codes, push notification,copy online list to offline list etc).

Testing Phase:

* Conduct unit testing.
* Perform some sort of QA.
* Optimize performance and fix bugs.
* Gather initial user feedback on UI/UX.
* Security testing, especially for user data and authentication processes.

Deployment Phase:

* Finalize app store deployment (iOS and Android).
* Provide internal "soft" versions to gather user feedback using tools such as TestFlight on iOS.
* Full-scale release and marketing.

**Key Features**

* List Creation & Organization: Users can create, categorize, and manage various lists (e.g., movies, restaurants, tasks).
* Collaboration: Share lists with friends or family, allowing for real-time edits and additions.
* Random Selection: Use "I Feel Lucky" or similar features to randomly select an entry from a list for decision-making.
* Progress Tracking: Track progress for completed tasks or items, and set priorities for each entry.
* Camera Integration: Leverage camera and QR code technology for quick collaboration on shared lists.
* Push Notifications: Send updates on shared lists.

**Constraints**

* Limited timeline for the initial release due to the length of the semester, targeting a 3-4 month development period.
* The project must remain within a fixed budget allocated for development, marketing, and testing.
* The app is being developed with cross-platform tools (probably React Native), which may limit some platform-specific capabilities.

**Dependencies**

* MongoDB as a database, storing data.
* Push Notification Services for Android and iOS.
* React Native Libraries
* External Stakeholders
* Beta testing and feedback collection from early users.

**Deliverables**

* Fully functional app for iOS and Android.
* Detailed built-in user tutorial on how to use the app, manage lists, and collaborate with others.
* Codebase and architecture documentation for future development and maintenance.
* Results from QA, including unit tests, and performance benchmarks.
* App descriptions, screenshots, and promotional content for app stores and online marketing.

**Timeline**

Design Phase: 2-3 weeks

Wireframing, UI/UX prototyping, and user flow design.

Development Phase: 8-10 weeks

Core features implementation, database setup, backend integration, and sensor functionality.

Testing Phase: 3-4 weeks

QA testing, bug fixing, user testing, and performance optimization.

Launch & Marketing: 2 weeks

App store submission, final bug fixes, and user acquisition campaign.

(Of course this wouldn't be the case for the final version that will be presented, but this is what we're roughly aiming to achieve.)

**Performance Criteria**

* The app must pass all unit, integration, and user acceptance tests.
* Aim for an initial user rating of at least 4.5 stars across both iOS and Android platforms.
* All key features (list creation, collaboration, random selection, push notifications) must be fully functional and bug-free.
* Achieve a minimum of 5,000 downloads in the first month after launch to motivate further marketing funds.

**Risk Identification**

* Possible technical challenges in integrating third-party APIs and frameworks, leading to delays.
* Limited team size or expertise in certain technologies could slow development.

**Mitigation Strategies**

* Perform early testing with third-party services to identify potential integration challenges.
* Hire talented and work-independent individuals which can easily adapt to the required technologies.

**Contingency Plans**

* Develop a backup plan using alternative stack of technologies.
* Delay certain features to later releases.
* Adjust timelines or completely cut features.

**FEASIBILITY STUDY REPORT**

**Technical Feasibility**

* Initial UI/UX will be defined in Figma
* Cross-platform development via React Native.
* Cloud Database via MongoDB Atlas.
* Backend via Node.js, Axios and Mongoose.
* Deployment via AWS.

**Operational Feasibility**

* Consolidates multiple list types (private, shared) into one intuitive platform.
* Enables groups to organize and share recommendations, streamlining decision-making for collective tasks (e.g. picking restaurants or planning activities).
* Interactive features such as progress tracking and random selection tools keep users motivated to use the app regularly.

**Financial Feasibility**

Our main financial model is 10$/year via a premium subscription for each user, targeting approx. 10,000 users during the first year, growing by ~20% annually. On the other hand, our team consists of a total of 3 Software Engineers, costing the company 100$/hour each. We estimate development to take a total of 2000 hours, based upon:

* **Design Phase**: 300 hours.
* **Development Phase**: 1,200 hours.
* **Testing Phase**: 400 hours.
* I**mplementation Phase**: 100 hours.

Which is approximately a single year of development. Combining all of this, we can gather our NPV:

* Total cost of Development: 200,000$, assuming each engineer works for exactly 666.7 hours.
* Revenue per premium user: 10$/year
* Targeting 10,000 premium users during the first year, growing by 20% annually.
* Discount rate of 8% (to account for the time value of money - money is worth more today than it will be in the future, as stated in lecture 2).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Year** | **User Base** | **Revenue** | **Discount Factor** | **Discounted Revenue** |
| **1** | **10,000** | **$100,000** | **0.93** | **$93,000** |
| **2** | **12,000** | **$120,000** | **0.86** | **$103,200** |
| **3** | **14,400** | **$144,000** | **0.79** | **$113,760** |
| **4** | **17,280** | **$172,800** | **0.73** | **$126,144** |
| **5** | **20,736** | **$207,360** | **0.68** | **$141,005** |

Total Discounted Revenue: $577,109.  
NPV = Total Discounted Revenue - Initial Investment:  
NPV = $577,109 - $200,000 = $377,109.

**Most importantly, the NPV is positive which means that the project is finacially feasible.**

**Return on Investment (ROI):**

Recall:ROI = (Net Gain / Cost of Investment) × 100  
Net Gain = Total Revenue - Initial Investment

The higher the ROI, the higher the potential financial return relative to investment.

Inputting our parameters, we get:

Net Gain = $744,160 - $200,000 = $544,160

ROI = ($544,160 / $200,000) × 100 = 272%.

**STAKEHOLDER ANALYSIS DOCUMENT**

**Stakeholder List**

* Development Team: Software Engineers, each with a vast skills and responsibilities. Responsible for development and deployment.
* End-Users: our market
* Project Sponsors: Potential Investors
* Marketing Team: Responsible for marketing.

**Interested and Expectations**

|  |  |  |
| --- | --- | --- |
| Stakeholder Role | Primary Interests | Expectations |
| Software Engineers | Full-on Development of the app. | Proper task allocation, technical clarity, and resources. |
| End-Users | Functional, easy-to-use app that adds value to their routines. | Reliability, collaboration features, and engaging design. |
| Investors/Project Sponsors | A profitable and market-competitive application. | Positive ROI, transparent reporting, and app scalability. |
| Marketing Team | Strong app engagement metrics and user growth. | High-quality content and user-friendly features. |

**Communication Plan**

|  |  |  |  |
| --- | --- | --- | --- |
| **Stakeholder Group** | **Communication Channel** | **Frequency** | **Purpose** |
| **Development Team** | Slack and Trello | Daily standups, weekly syncs | Monitor progress, address blockers, and task updates. |
| **End-Users** | Email newsletters, app notifications, app store | Bi-weekly during beta, monthly post-launch | Gather feedback, promote features, and drive engagement. |
| **Project Sponsors** | Reports, dashboards, and Zoom calls | Bi-weekly | Share project status, budget updates, and risks. |
| **Marketing Team** | Google Drive for assets, Zoom calls | Weekly | Align promotional strategies and updates on development progress. |