**a. Revise your System Request & Feasibility Analysis**

*(Note: The previous assignments, the System Request and Feasibility Analysis, seem complete and meet the requirements based on the rubric.)*

**b.i. How We'll Figure Out What's Needed**

Besides just interviewing people, we need a couple of other ways to gather requirements for the NeighborShare project. We'll use Document Analysis and Questionnaires.

**1. Looking at Existing Documents**  
This just means we'll read through documents that already exist to find any rules or goals we need to know about. For this project, we'd look at:

* **HOA Rules:** We need to get the official bylaws from a few different HOAs. This is important because they might have rules about lending things out, especially if money is involved. We need to make sure our app doesn't break any of their rules.
* **City Plans:** Since the city is a sponsor, we should look at their "Smart City" or "Green Initiative" plans. This will help us use the same words they do when we talk about the project's benefits, like how it helps reduce waste. It makes sure we're on the same page as them.

**2. Questionnaires**  
To get info from a lot of people quickly, we'll make a simple online survey. We can send it out through the city's email list or post it on community Facebook pages.

* **What it will do:** The survey will help us check if our guesses are right and get some real numbers. We can ask things like:
  + What kind of stuff have you needed to borrow in the past year? (With a checklist: Power Tools, Camping Gear, etc.)
  + What’s your biggest worry about sharing? (Let them rank stuff like item damage, theft, or scheduling problems.)
  + What’s a fair service fee for an app that handles all the annoying parts for you? This will give us a starting point for the business side of things.

**b.ii. How We'll Analyze the Info**

Once we have all this information, we need to make sense of it. We'll use two main approaches.

**1. Finding the Real Problem (Root Cause Analysis)**  
This is a way to dig deeper than the obvious problem. We want to make sure we're fixing the right thing.

* **How we'll do it:** The problem we see is "Using Facebook to borrow stuff is unreliable." So we ask why.
  + **Problem:** People ignore messages on Facebook.
  + **Why?** Because there's nothing holding them accountable.
  + **Why?** Because it's just a social media site, not a system for transactions. It doesn't have calendars, payments, ID checks, or reviews.
  + **What this tells us:** The real problem isn't just that the sites are "clunky." It's that they lack Trust and Structure. This proves that features like ID verification, deposits, and user ratings aren't just "nice to have," they are essential.

**2. Checking Out the Competition (Informal Benchmarking)**  
This means we'll look at other apps and services to see what they do right and wrong.

* **How we'll do it:** We'll compare our NeighborShare idea to what people use now: Facebook, Nextdoor, and maybe a real sharing app like Peerby. We can even look at Airbnb because they're great at making people trust sharing valuable property.
  + We'll make a chart to see who does what: Does Nextdoor verify users? (No). Does Airbnb? (Yes). Does Facebook handle payments? (No).
  + **What this tells us:** We'll see exactly what features will make our app better. The main weakness of the current methods is the lack of a system for trust. Airbnb shows us that handling payments and deposits is key. This gives us a clear list of features we need to build to be successful.

**b.iii. Interview Questions**

**Stakeholder:** A regular resident in the neighborhood

**Open-Ended Questions:**

1. Can you tell me about a time you needed a tool for a project? How did ya end up getting it?
2. Do you talk to your neighbors much now? What would make you feel more connected to them?
3. What's the one thing that would make you nervous about lending something expensive to a neighbor?
4. If you could wave a magic wand and create the perfect app for sharing stuff, what would it do?
5. What would you want to happen if you lent something out and it came back with a scratch on it?

**Closed-Ended Questions:**

1. In the last year, did you buy a tool you only used once? (Yes/No)
2. On a scale of 1 to 5, how important is it that the app confirms everyone is who they say they are?
3. Would you pay a small fee, like a couple of bucks, if the app handled the deposit and insurance for you? (Yes/No)
4. Which of these would you be most likely to borrow? (Tools, Camping Gear, Kitchen Stuff, Party Supplies)
5. Would you use this more on your phone or on a computer?

**c. 10 Functional Requirements**

1. The system needs to let users sign up with an email and password.
2. A user has to be able to upload their ID to get verified.
3. Users should be able to list their items with pictures, a description, and a price.
4. There should be a calendar so people can show when their stuff is available.
5. The app needs a search bar so people can find things by name, date, or location.
6. There has to be a messaging system so the lender and borrower can chat.
7. The system should handle the whole booking process: request, accept/decline.
8. It must be able to handle payments for fees and deposits securely (using something like Stripe or PayPal).
9. The app needs to send automatic reminders before pickup and return times.
10. After a rental is done, both people should be able to rate each other and the item.

**d. Non-Functional Requirements**

**i. Operational Requirements**

1. The website has to work on Chrome, Firefox, and Safari….
2. The mobile app needs to be on the Apple App Store and Google Play Store.
3. We'll host it on a cloud server like AWS to ensure scalability and reliability..
4. The system should back itself up every day automatically.

**ii. Performance Requirements**

1. Search results should show up in less than 2 seconds.
2. The app needs to be online and working pretty much all the time (99.9% uptime).
3. It should work fine even if a lot of people are using it at once.
4. Item pages with lots of pictures should load fast.

**iii. Security Requirements**

1. All passwords have to be stored securely using hashing and salting.
2. The system has to follow all the standard rules for handling credit cards (PCI DSS).
3. All communication between the app and the server needs to be encrypted (TLS).
4. Personal info, like driver's licenses, must be kept super secure and deleted after being used.

**iv. Cultural & Political Requirements**

1. The system has to follow Canada's privacy laws (PIPEDA).
2. Users must agree to a "Code of Conduct" so everyone is respectful.
3. The app's design should be easy for people with disabilities to use (follow WCAG rules).
4. There needs to be an easy way to report problem users or disputes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MGMT 1223 - Systems Development 1 - Assignment #4** | | | | |
|  | **Team Member Names:** |  |  |
|  | 3 = Yes, 2 = Somewhat, 1 = Barely, 0 = No | Self | Prof |
|  | **Totals** | 0 | 0 |
| **Submission Logistics (Yes or No)** |  |  |  |
|  | Was the rubric submitted with partner names and self-assessment completed? | 3 |  |
|  | Was this report included as an addition to the preceding system request and feasibility analysis? | 3 |  |
|  | Do the combined elements cohesively describe the project? | 3 |  |
|  | Was the analysis updated in response to professor feedback (if necessary)? | 3 |  |
|  |  |  |  |
| **Requirements Approach #1** |  |  |  |
|  | Is the chosen approach a good choice for the problem being worked on? | 3 |  |
|  | Is the description of how the approach will/can be used accurate and reasonable in the context of the problem being worked on? | 3 |  |
|  |  |  |  |
| **Requirements Approach #2** |  |  |  |
|  | Is the chosen approach a good choice for the problem being worked on? | 3 |  |
|  | Is the description of how the approach will/can be used accurate and reasonable in the context of the problem being worked on? | 3 |  |
|  |  |  |  |
| **Analysis Approach #1** |  |  |  |
|  | Is the chosen analysis strategy a good choice for the problem being worked on? | 3 |  |
|  | Is the description of how the approach will/can be used accurate and reasonable in the context of the problem being worked on? | 3 |  |
|  |  |  |  |
| **Analysis Approach #2** |  |  |  |
|  | Is the chosen analysis strategy a good choice for the problem being worked on? | 3 |  |
|  | Is the description of how the approach will/can be used accurate and reasonable in the context of the problem being worked on? | 3 |  |
|  |  |  |  |
| **Interview Questions (2 marks each)** |  |  |  |
|  | Are the open-ended questions likely to provide meanigful information to inform your analysis? | 3 |  |
|  | Are the closed-ended questions likely to provide meanigful information to inform your analysis? | 3 |  |
|  |  |  |  |
| **Functional Requirements (2 marks each)** |  |  |  |
|  | Do the functional requirements effectively describe either a process the system must perform or information it must contain? | 3 |  |
|  |  |  |  |
| **Non-Functional Requirements (2 marks each)** |  |  |  |
|  | Do the operational requirements effectively describe the operating environment in which the system must perform? | 3 |  |
|  | Do the stated performance requirements effectively describe issues regarding response times, capacity or reliability? | 3 |  |
|  | Do the stated security requirements effectively describe system features to protect the security of the proposed system? | 3 |  |
|  | Do the stated cultural and political requirements effectively address issues specific to the country or countries in which the system will be used? | 3 |  |