

Personalized Shopping Assistant

Team 4

Thomas Ripp

Joseph Mirabile

Jose Cruz

Stevens Institute of Technology

April 30, 2021

Contents

| | | |
|----------|--|-----------|
| 1 | Elicitation | 2 |
| 1.1 | Context | 2 |
| 1.2 | Methods | 2 |
| 1.2.1 | Brainstorming | 2 |
| 1.2.2 | Interviews | 3 |
| 2 | Analysis | 4 |
| 2.1 | Mission Statement, Key Drivers, Key Constraints | 4 |
| 2.2 | Key Stakeholders, User Requirements, Business Requirements . . | 5 |
| 2.3 | System Requirements, Security Requirements, Quality Requirements | 6 |
| 3 | Specification | 8 |
| 4 | Validation | 10 |
| 5 | Appendix | 12 |

1 Elicitation

1.1 Context

For the Personalized Shopping Assistant project we set up meetings every other week with our customers, because of COVID we had to have this meeting in a remote setting using Google Meet.

1.2 Methods

1.2.1 Brainstorming

For the first meeting, our goal as a team was to figure out what is the main goal of the project, identify the stakeholders and get an idea of the scope of the project. For this project we use user personas [1] to identify the stakeholders, because the customer do not have end users at hand so we have to create imaginary roles as reference.

Next we wanted to understand the customers background, so we know how to translate ideas using the customers language as reference, this background helps us to create analogies that the client can understand.

To achieve this we set up the meeting in a way that each of the people involved in the project introduce themselves and tell us what kind of experience they have.

Once we establish this, we have a baseline of the technical vocabulary that we can use during the project.

We noticed the following by observing the group dynamic:

- Samantha and Timothy focus mainly on the business side, they are the ones that guide the conversation and have a better understanding on how the application should behave if a particular edge case appears.
- Alex focus more towards user privacy, security and techinal implementation.
- Kristin more towards user experience, on how to make the application more approachable towards the users.

We discover this by enabling them to talk freely about what is the goal of the project, and each of them tends to participate when a particular subject is discussed.

1.2.2 Interviews

From the second meeting onwards, we take the input from the previous meeting and we create a set of questions that we send to the client 2-3 days before the next meeting so they have time to discuss.

We use this to get some clarification about requirements that were too vague or to confirm how a particular edge case should be handled.

These questions help us to stay on scope during the meeting, once we are in the meeting, we review the questions one by one and ask more information for clarification, and depending on the use case we move the conversations towards how to validate them or what quality attributes they would expect from the use case.

If the client does not have the answer to a particular question we skip that question and go to the next one. Once we finish with the questions, we backtrack to the questions that they did not have an answer and we start brainstorming about possible solutions.

While we are doing the brainstorming session we validate right away, how can we verify those requirements and also we evaluate if those are feasible or not, sometimes in the conversations, some edge cases arise and we think about possible solutions for those edge cases.

When the meeting finalizes, we take notes of all the answers and we start analysing them.

2 Analysis

After each meeting we review the notes, try to reply the following questions.

1. Are there any new requirements?
2. The new requirements conflicts with existing one?
3. What kind of requirements are this new requirements?
4. Do we need to update any existing requirements?
5. Do we know all the possible edge cases?
6. Do we know how to solve the edge cases?

2.1 Mission Statement, Key Drivers, Key Constraints

The goal for the meeting was to determine:

1. What is their motivation to create this project?
2. What problem in the market do they want to fulfill?
3. How do they want to achieve this?
4. What is the business model?
5. How big is the market for this project, short and long term?
6. Who are their competitors?
7. What is their product differentiator?
8. What kind of constraints do they have, either time, technological or monetary?

We had a meeting in a free form setting, brainstorm, so the customers could provide their insight without any specific agenda, sometimes one or more of the people in the group focused on an area in particular and we moved the conversation over that topic.

2.2 Key Stakeholders, User Requirements, Business Requirements

Once we established why the client would like to develop this project we went ahead and tried to figure out who are the stakeholders, what is the scope of the project and what business goals are they trying to achieve.

The goal for the meeting was to determine:

1. How many stakeholders are in the project?
2. What is the stakeholder main motivator?
3. How the project would help the stakeholders?
4. What are their business goals in the short, medium and long term?
5. How many systems do we need to build for this project?
6. What functionalities do each stakeholder want?
7. Seize what is the scale on which the program is going to run in the short, medium and long term
8. What is their budget?
9. What is their go to market timeline? *To establish a deadline.*

We sent the clients a document with the questions that we had from the previous meeting, some were to make sure that we understood their motivation, others were to get clarification because some of the replies seemed ambiguous.

We used an interview approach to get these clarifications, this process was quick so we moved the conversation towards getting to know what is their current status regarding resources and constraints, what is their budget, what is their business goals, how much traffic they were expecting.

We use this information to create their business requirements. Once we have these we start talking about the key stakeholders for the program.

1. What was their motivation?
2. What is their target audience? *Demographic*
3. What is their social-economic status?
4. How are they solving their problem right now, and how the project would improve the existing solution.

We move from a interview format towards brainstorming, we want to open the scope as much as possible to have place for creativity and during the conversation we start closing the scope by getting into more details.

We achieve this by discarding ideas that were not feasible, either from a technological perspective or time constraints.

With this we identify the key stakeholders, and then we start discussing what kind of functionalities each of these stakeholders would want to achieve with the program, this was still in a brainstorming format.

We used this information to generate the User Requirements.

2.3 System Requirements, Security Requirements, Quality Requirements

With the information from the previous meeting some problem arises, there was conflict in some of their requirements, we take notices of those and we put them in a document that we sent before the meeting.

The goal for the meeting was:

1. Clarification of the conflicting requirements
2. Use case validation
3. Quality attributes
 - Performance
 - Accessibility
 - Usability
 - Security
4. How many platforms are going to be develop?
5. What kind of constraints do they have regarding providers?
6. What kind of compliance the software requires?

From the previous meeting we know what is the goal and motivation for the stakeholders, and also how they interact with one another, now the focus is moving towards how can we achieve those expectations using software.

We as a team already have an idea on how the project could be implemented and which providers to use to achieve it.

We were giving suggestions on how the system could be implemented.

Now it is time to transform these user requirements into system requirements, we start asking them questions about what the process should be and ask them on how the system should behave in case the system reaches a particular edge case.

During the conversation we ask questions on how to validate if this use case was executed successfully, these were the foundations for the quality requirements.

Once we understood how can we verify if the requirements were successful or not, we move the conversations towards how good this were supposed to be, we start asking questions about:

- How fast X should be able to perform
- How many steps should happen to achieve a particular use case?
- How should we do data validation?
- What is the expected uptime for a particular system?
- When are the expected peak hours, if any?

We take these notes and create a formal representation of these requirements using the grammar used in the textbook.

3 Specification

From the conversation that we have with our customers, they have particular keywords that they use that help determine what kind of requirements they are talking about.

If they use terms like market share, revenue, sales or terms that are associated with money, we know that they are business requirements.

If the sentences include a functionality or a feature or something that the system should be able to do is a **user requirement**, and during the meeting we try to decompose those into what are the inputs and the expected outputs of that functionality so we can use them later for validation and to write the system requirements properly.

When we are talking about a particular feature, if the customer specifies how it should be a particular function we categorize those as **functional requirements**.

In this project they have two main focuses, regarding quality, these are **speed** and **security**, anytime they mention the word fast, or private data, we know that they are talking about quality attributes.

During the interviews we try to push them to figure out the system edge cases, and also we ask them to imagine how the application should work under certain conditions, this would be our **constraints**.

To figure out how many systems are we building, we use the previous exercise to imagine how the application would work, the most frequent words they use to describe the system is browser and apps, so we ask for clarification on which browser, *Mobile or Desktop*, and which platform, *iOS or Android*, are they talking about, this ones would be ours **external interface requirements**.

In general, during the meeting anytime one of the customers uses words like *standard* or *normal* we ask them to clarify those to avoid ambiguity.

When they use terms like, *perhaps*, *maybe*, *could*, *sometimes*, we move the conversation in a way that they can evaluate all the possible cases and define what is the precise requirements.

If they are not able to articulate it, we take notes of that as a comments and we tell them that we can't add that into the system unless they define how the system should behave.

If they provide some type of metric either distance or time, we ask them to specify the unit and the values, this are the **performance requirements**

If they offer a solution to a problem or they make a feature request, we ask them:

1. How that solution or feature would help the user?
2. How the implementation would work?
3. What would be the required steps?
4. How the system should managed each edge cases?
5. Do we need to track new data for this feature?
6. Does that feature affects existing requirements?

Sometimes the solution is too ambiguous and the customers decide that they do not understand the solution well enough to be added it to the requirements, other times that solution creates new requirements and also changes existing ones.

4 Validation

At this point in time we have a list of all requirements that the client would like to see in the project. We do a review of the list and we start discarding the requirements that are not feasible, because of technical, geographical or monetary reasons.

Since this project is a CRUD application [3] and do not require anything special from the hardware besides GPS and internet connection, most of the requirements are feasible, with the exception of some of performance requirements, were the client requested 15ms on page load.

Latency is an issue in distributed systems, making the requirements infeasible because of physics laws.

Next we ask them to put each system and requirement into the following categories:

- **MVP** [2]
- **Future Release**

For the **MVP** we ask them which features and systems are required for their release date, we ask them to be as objective as possible. Anything that is not in the **MVP** is going to **Future Release**.

We focus the conversation over the MVP, now that we know which features and systems are important we need to prioritize them, we send the client a document with all the features, grouped by system, and ask them to rank them between 1 and 5.. *1 being the highest priority.*

We use **dialog maps** to validate the workflow for each of the systems, the dialog maps are available in the Appendix.

- For the **Personal Shopper System** we used the Figure 1.
- For the **Personal Shopper Driver System** we used the Figure 2.
- For the **Personal Shopper Merchant System** we used the Figure 3.

References

- [1] Patrick Faller 2019, Dec 17, Putting Personas to Work in UX Design: What They Are and Why They're Important <https://xd.adobe.com/ideas/process/user-research/putting-personas-to-work-in-ux-design/>
- [2] ProductPlan 2021, April 30, Minimum Viable Product (MVP) <https://www.productplan.com/glossary/minimum-viable-product/>
- [3] Brenden Thornton 2019, Aug 30, What's a CRUD App? <https://medium.com/@thorntonbrenden/whats-a-crud-app-e5a29cce03b5>

5 Appendix

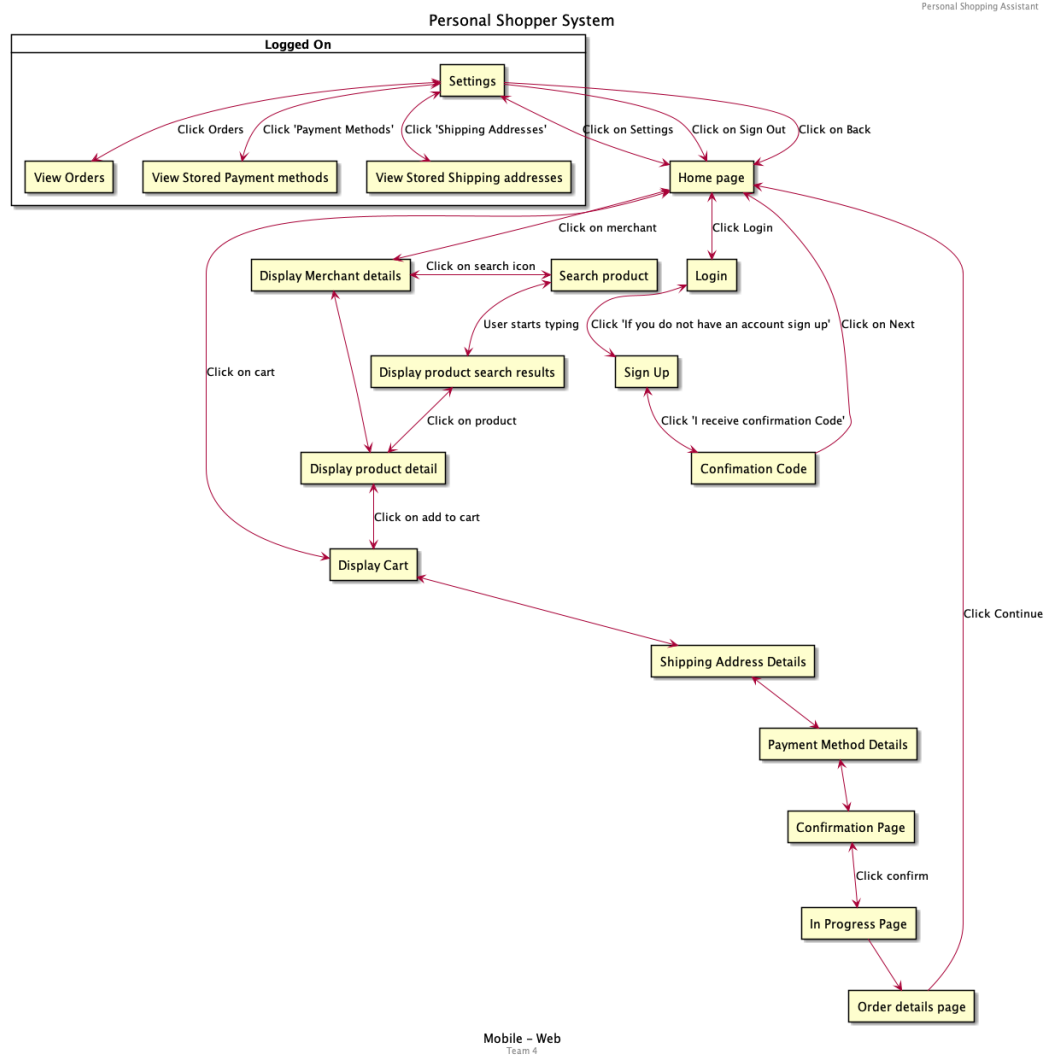


Figure 1: PSS Dialog Map

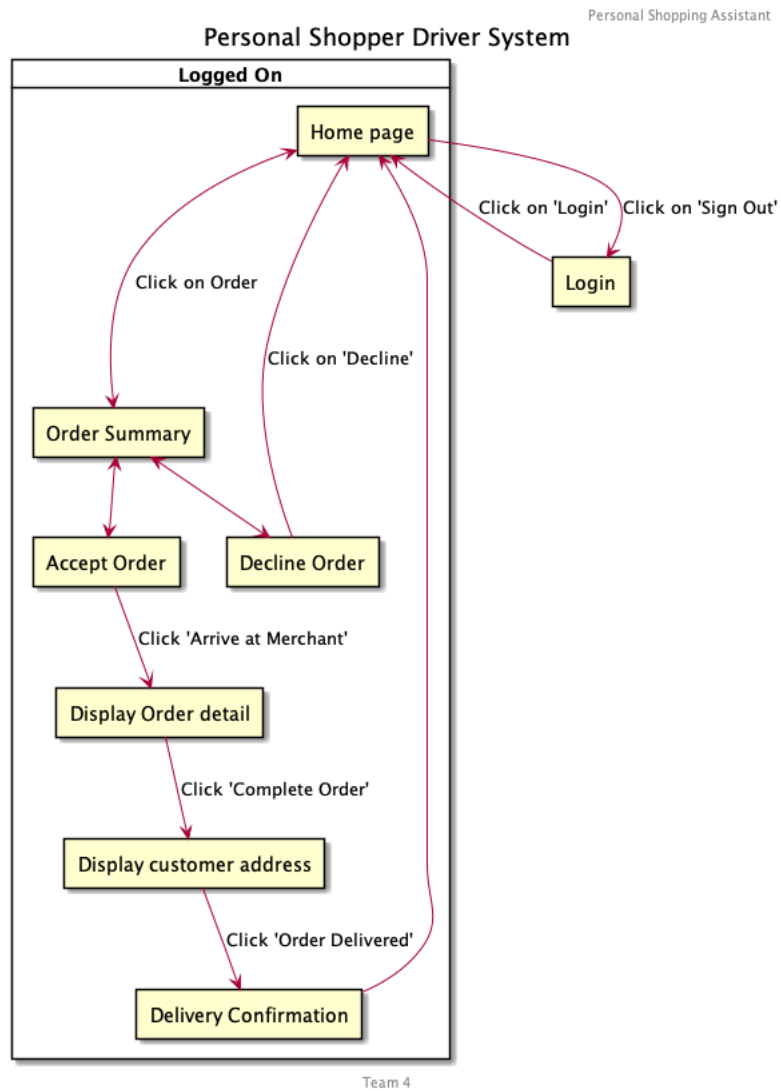


Figure 2: PSDS Dialog Map

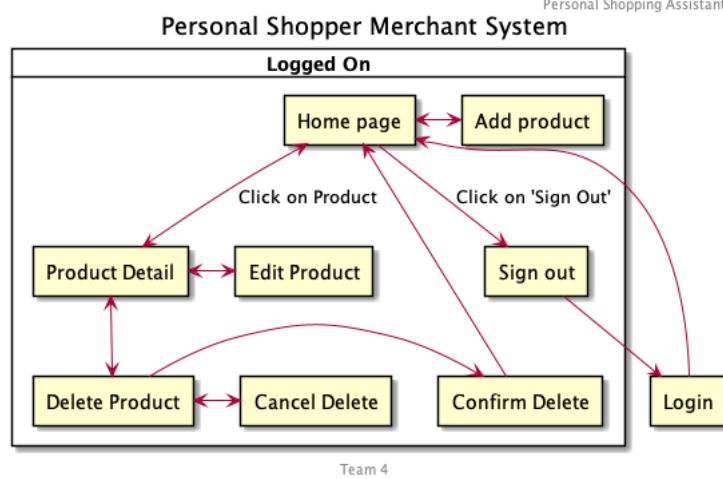


Figure 3: PSMS Dialog Map