Requirement Specification

Målutgivelse	
Epic	
Dokumentstatus	UTKAST
Dokumenteier	Fredrik Waaler
Designer	Klaus Dyvik
Utviklere	
Kvalitetssikring	

Goals

Background and strategic fit

"Sukker på Toppen" is a new cake design factory in Ålesund, which aims for remarkable cakes for special occations. Its owner, Lais Gomes Marcolongo ac cumulates all the functions in the company, including designing, baking, social media management and accounting. With so many responsibilities, automatizing repetitive and time-consuming tasks can go a long way. Therefore, the owner wants to minimize the workload on accounting tasks by automating them. The idea covers making an image recognizing system for analyzing receipts and invoices and sending this to Fiken, so the owner do not need to manually type in all information herself.

Assumptions

- Only one user is going to use the software
- The user has an android phone
- The user uses Fiken
- The user uses iZettle

Functional Requirements

#	Title	User Story	Importance	Notes	Measurements
1	Web Application	The user should be able to enter a URL in their browser that provides a tool for interacting with the service.	Must Have	▪ Create webserver	Webserver is up and running
2	File upload	The user should be able to upload a picture and/or PDF of an invoice or receipt. It should be sent to server for processing.	Must Have	▪ Using the website.	This can be verified by showing the picture/pdf in the web application and manually comparing the received and the selected picture.
3	Server image processing	When uploaded via the website, the invoice/receipt should be processed by the server and the correct data should be extracted and presented to the user.	Must Have	▪ Google Vision will extract the text, and serverside processing will present the user with a proposed ledger entry. The server must be able to sort appropriately by MVA, customer /salesman and product category.	This can be verified by checking that the appropriate text is placed with the appropriate labels when a ledger entry is proposed.
4	Upload to fikens ledger	The server should send the user-approved ledger entry to the users Fiken account.	Must Have	▪ Using Fikens API	We should be able to verify that the data arrives at Fiken the way the user approved it.
5	Database storage	The user should be able to log in using email and password. For authentication purposes this needs to be stored in a database. The database should also provide intermediary storage for pictures submitted from the app, until the user approves and submits to Fiken.	Must Have	▪ Fiken username and password must also be stored to enable interaction with Fikens API.	We should be able to fetch all relevant data from the database.

6	Editing of processed business data	The user should be able to edit the processed business data before it is sent to Fikens ledger.	Must Have	▪ Using the the website. This is the data that will be sent to Fikens ledger.	We should be able to see that the edited data is being accepted and verified by Fikens ledger.
7	View previous ledger submissions	The user should be able to view previous ledger submissions and filter by various flags (seller, date, category etc.).	Nice to have	▪ Using the website. To get a quick overview of previous submissions.	Show a table/log of all submissions. The submission can be ordered by date, price or seller.
8	Mobile Application for submitting images	The user should be able to take a picture of the invoice/receipt to submit it to the server for processing through an Android app.	Nice to have	▪ This is an alternative (not replacing) to the web-based application.	Successful if a image processed via the app creates a correct ledger entry in Fiken.

Non-Functional Requirements

#	Title	User Story	Importance	Notes	Measurements
1	Security	The system should be able to control the access and session to verify if the user is still logged in and valid. Passwords and emails should be stored in a secure format in the database. Secure communication has to be used to transfer data.	Must Have		If the user is inactive for x minutes, auto log out. Given full database access, it should not be possible to crack passwords and emails. Uses HTTPS.
2	Concurrency and capacity	The application is sequential, and the number of concurrent users are always one. The system can store up to 25 GB of data on the server.	Must Have		The user has all available storage at disposal.
3	Performance	The service should not interrupt normal workflow while waiting for processing.	Must Have		Should not stall the UI for longer than 5 seconds.
4	Reliability	Notify the user whenever the application performs an action on behalf on the user.	Nice to have		A new user should be able to sit through and understand what happens.
5	Maintainability	Maintenance should not shut down the system. Minimal amount of maintenance should be needed. System should always run.	Must Have		
6	Usability	Assuming the user is familiar with Fiken, the application should be intuitive and easy to interact with.	Must Have		A new user should independently understand how to use the application.
7	Documentation	The project should be thoroughly documented in the source code, alongside other documentation.	Must Have		The project will be delivered with extended documentation.

Questions

Nedenfor er en liste over spørsmål som skal tas stilling til som følge av dette kravdokumentet:

Question / problems	Solution
What to do if there is no connection to the server?	Inform the user and wait til connection is established.
What if the servers storage is full?	Either inform the user and await manual deletion of old ledger entries, or implement a basic solution (for instance deleting the oldest one).

Sources

https://startxconsulting.com/en/articles/7-non-functional-requirements-you-should-always-keep-into-account/?fbclid=IwAR2EVQ9PjklPgyyTq2yvOhM9TDFlgY-CYdyxJQ5j-bV2eySDeJxudFSLtq0