Final Project in Software Engineering

Application Requirements Document

Omer Presler, Amit Grumet, Rotem Cohen

https://www.canva.com/design/DAFT6ZpR8Ek/xVr8omszBJsy1dYqDdpZGA/view?utm_content=DAFT6ZpR8Ek&utm_campaign=designshare&utm_medium=link&utm_source=publishsharelink

Chapter 1 - Introduction

1.1 The Problem Domain

Kibbutz Maabarot in Emek Hefer is a cooperative kibbutz. All its businesses are managed in the kibbutz and provide services for the kibbutz members (and for the kibbutz members only). Today each business works with a different system that provides services for the members. This is inconvenient since there must be a significant human involvement when transferring the information on expenses to the accounting. In addition, the current systems don't supply the best answer to the business's services. The maintenance business, which we start focusing on, has a system that is used by the kibbutz members. A member that has a malfunction in his home, can request a repair from the maintenance. His request has two statuses- "open" and "closed", although there are more possible statuses such as "received by the system", "read by a repairman", "in progress", "will be repaired by tomorrow" and more.

1.2 Vision

We are planning on creating a well-fitted system for kibbutz businesses. At first, we plan on focusing on the maintenance of the kibbutz in addition to making a "Kibbutz cash" register for each branch in the kibbutz, and later expand the use of our system to other businesses of the kibbutz. The expansion should be simpler than the first process of working with one business since we can convert the request of a repair, for example, to requesting a different service in another business, and all those systems will assemble one large system that will serve the kibbutz members.

1.3 Stakeholders

This project is for our customer kibbutz Maabarot. We are planning a system that will assist the kibbutz businesses to work more efficiently, with less human intervention. To be more specific, at first, we will focus on the maintenance of the kibbutz. We will understand the needs of the system, and later we will proceed to work for other businesses, such as the bakery, laundry, hairdresser, and so on…

1.4 Software Context

The project is creating a client-server website, with a database of requests, providing services, and item/worker management. The program will have multiple windows in which the “clients” can use the system. The “service provider window” will provide a way for the business to manage its own workers, requests, and services they want to provide, add photos and massages to the page, etc, on the kibbutz member side the landing page will provide a way to ask the store for something and start a request. The “system manager” window will let any administrator see the full flow of the system and will provide access to individual services data, the administrator can see requests, see the stock status, and can intervene in requests, i.e., cancel requests, edit requests, etc. In addition, whenever needed the business can produce a sheet with all its requests and purchases in x time frame, it could be a year, a month, or a week and provide important stats. The business can provide all that have been “bought” in that time frame, then that sheet can be automatically sent to accounting. The “accounting” window will have access to business data, they will be able to produce said sheets.

Chapter 2 - Usage Scenarios

2.1 User Profiles — The Actors

1) Admin: system administrator is a user with permission to alter the database and the system's businesses and add new users to the system.

2) Business: a business in the kibbutz

3) Kibbutz member: any kibbutz member, who can use the services of the kibbutz's businesses.

4) Accounting: accounting users can get business expenses and profit reports.

2.2 Use-cases

1) Add a new user:

Actor: Admin

Pre-condition: None

Post-condition: A user is added to the users' database.

Main-flow: An admin enters a user's details: name, budget number (if has any, the user may not be a kibbutz member, but only a business worker), phone number, and email address, then clicks on "Add", and the new user is added to the users' database.

Alternative-flow: When one of the details is incorrect; a phone number with more or less than 10 digits or one of the mandatory details is missing, the user will not be added to the database, and feedback for failed action displays to the admin's screen.

Positive test: After adding a new user, check if he exists in the users' database.

Negative test: After failed adding action, check the user wasn't added to the users' database.

2) Notification when request status change:

Actor: Business

Pre-condition: A request exist and can be altered

Post-condition: a notification has occurred

Main-flow: One of the stakeholders in that request has changed something, the system recognizes a change and notifies whoever needs to be notified.

Alternative-flow: None

Positive test: After altering a request a notification will occur.

Negative test: None

3) Notification when requests are created:

Actor: Business

Pre-condition: A request can be made

Post-condition: A notification has occurred

Main-flow: A member requests something new, and the system recognizes a change and notifies whoever needs to be notified.

Alternative-flow: None

Positive test: After creating a new request, a notification will occur.

Negative test: None

4) New request created:

Actor: Business

Pre-condition: A request can be made

Post-condition: A request being created

Main-flow: A member opens a business, browses, adds what he wants, and requests it (request cakes or services whatever is offered in that store) after that a request is being created.

Alternative-flow: None

Positive test: After creating a new request a new request will appear.

Negative test: After creating an invalid new request no new request will appear.

5) On-Site Purchase:

Actor: Kibbutz member, Business

Pre-condition: Member's account isn't locked.

Post-condition: The purchase is saved in the purchases database

Main-flow: A member buys a product from a business, and the cashier adds the purchase to the database through the system and thus concludes the transaction.

Alternative-flow: None

Positive test: After creating a new purchaseת the purchase will appear in the database.

Negative test: After creating an invalid request no new request will appear in the database.

6) Add a product to a business website:

Actor: Business

Pre-condition: The actor has the permission to add a product.

Post-condition: The product has been added to the store.

Main-flow: The actor enters the business, clicks on "Add new product", add a new product to the business adds photos.

Alternative-flow: When the product's details are incorrect, the product won't be added to the business, and an error message will display on the actor's screen.

Positive test: The product will be saved in the database after creating a new product.

Negative test: After creating an invalid product, the product will not be added to the database and an error message will appear.

7) Add a service to a business:

Actor: Business

Pre-condition: The actor has the permission to add a service.

Post-condition: The service is added to the business.

Main-flow: The actor enters the business, clicks on "Add new service", add a new service to the business.

Alternative-flow: When the service's details are incorrect, the product won't be added to the business, and an error message will display on the actor's screen.

Positive test: After creating a new service the product will be saved on the database.

Negative test: After creating an invalid service the product will not be added, and an error message will appear.

8) Remove a service from a business:

Actor: Business

Pre-condition: The actor has the permission to delete a service.

Post-condition: The service was deleted from the business.

Main-flow: The actor enters the business, enters the services list, picks the service he would like to remove, clicks on “Remove”, and the service is removed from the database.

Alternative-flow: None

Positive test: After removing an existing service the service will not be on the database.

Negative test: After removing a non existing service the service will not be removed, and an error message will appear.

9) Add business permission:

Actor: Business

Pre-condition: The user, to whom the business manager would like to add permission to, is a business worker.

Post-condition: The worker has the desired permission.

Main-flow: A business manager enters the business, clicks on “Business workers”, pick the worker, clicks on “Add permission”, and pick the permission (“Add stock permission”, “Add receiving requests permission”, etc.)

Alternative-flow: None

Positive test: After adding permission, the worker can make action according to the given permission.

Negative test: None

10) Remove a user:

Actor: Admin

Pre-condition: The user whom the admin wants to remove exists in the users' database.

Post-condition: The user was removed from the users' database and added to the non-active users' database.

Main-flow: An admin enters the list of all users, picks a user whom he would like to remove, clicks on "Remove", then the user is removed from the users' database, and added to the non-active users' database. We would like to maintain a database of non-active users so that we can restore a user if needed, and the admin and accounting can still review that user's actions and expenses.

Alternative-flow: None

Positive test: After removing a user, check if the databases are updated according to the post-condition.

Negative test: None

11) Restore a user:

Actor: Admin

Pre-condition: The user, whom the admin wants to restore, exists in the non-active users' database.

Post-condition: The user is removed from the non-active users' database and is added to the users' database.

Main-flow: An admin enters the non-active users' list, picks a user whom he would like to restore, clicks on "Restore", then this user is removed from the non-active users' database, and is added to the users' database.

Alternative-flow: None

Positive test: After restoring a user, check if the databases are updated according to the post-condition.

Negative test: None

12) Add new business:

Actor: Admin

Pre-condition: The new business's manager is a user in the system.

Post-condition: A business is added to the system.

Main-flow: An admin clicks on "Add new business", adds the business's details: name, business manager (from users' list), clicks on "Add", then the new business is added to the businesses' database.

Alternative-flow: If one of the details is incorrect, the business will not be added to the database, and feedback for failed action displays to the admin's screen.

Positive test: After adding, the businesses' database contains the business.

Negative test: After failed adding action, check that the businesses' database doesn't contain the business.

13) Remove business:

Actor: Admin

Pre-condition: The business which the admin wants to remove exists in the system.

Post-condition: The business was removed from the businesses' database, and was added to the non-active businesses' database.

Main-flow: An admin enters the businesses' list, picks a business which he would like to remove, clicks on "Remove", then removes the business from the businesses' database and adds it to the non-active businesses' database.

Alternative-flow: None

Positive test: After removing a business, check if the databases are updated according to the post-condition.

Negative test: None

14) Restore business:

Actor: Admin

Pre-condition: The business which the admin wants to restore exists in non-active businesses' database.

Post-condition: The business exists in the businesses' database and doesn't exist in the non-active database.

Main-flow: An admin enters the non-active businesses' list, picks a business which he would like to restore, clicks on "Restore", then the business is removed from the non-active businesses' database, and is added to businesses' the database.

Alternative-flow: None

Positive test: After restoring, check if the databases are updated according to the post-condition.

Negative test: None

15) Add new admin:

Actor: Admin

Pre-condition: The user, who the admin would like to make admin, is in the system.

Post-condition: The chosen user has admin permissions.

Main-flow: An admin enters the users' list, chooses a user, clicks on "Make admin", then the user has admin permissions.

Alternative-flow: None

Positive test: After adding admin, check if the user has admin permissions.

Negative test: None

16) Add new business worker:

Actor: Admin/ business manager

Pre-condition: The user, who the actor would like to make a business worker, is in the system.

Post-condition: The user is a business worker.

Main-flow: An actor enters the business, clicks on “Add new worker”, picks one from the users’ list, and clicks on “Add”.

Alternative-flow: None

Positive test: After adding a user as a business worker, the business manager can add business permissions to him.

Negative test: None

17) Remove business a worker:

Actor: Admin/ business

Pre-condition: The user, whom the actor would like to remove as a business worker, is a business worker.

Post-condition: The user is no longer a business worker.

Main-flow: An actor enters the business, clicks on “business workers”, picks the worker he would like to remove, and clicks “Remove”.

Alternative-flow: None

Positive test: After the removal, check if the user is no longer a business worker.

Negative test: None

18) Remove a product from a business:

Actor: Business

Pre-condition: The actor has the permission to delete a product.

Post-condition: The product was removed from the business.

Main-flow: The actor enters the business, enters the products' list, picks the product he would like to remove, clicks on "Remove", and the product is removed from the database.

Alternative-flow: None

Positive test: After removing an existing product the product will not be on the database.

Negative test: After removing a non existing product the product will not be deleted and an error message will appear.

19) Change request status:

Actor: Business

Pre-condition: There is a request in the business.

Post-condition: The request's status has been changed, and a notification is sent to the requesting member.

Main flow: The actor enters the request, clicks on "change status", changes it, and a notification is sent to the requesting member.

Alternative-flow: None

Positive test: After changing the status, the change has been made in the worker's site and in the requesting member's site.

Negative test: None

20) Close a request:

Actor: Business

Pre-condition: There is a request in the business.

Post-condition: The request appears as closed.

Main-flow: The actor enters the business, picks the request, changes its status to "Closed", and then the request moves from the open requests list to the closed requests list.

Alternative-flow: None

Positive test: After closing a request it will be moved to the closed requests list and be removed from the open requests list.

Negative test: None

21) Reject a request

Actor: Business

Pre-condition: There is a request to reject/accept.

Post-condition: The request appears as rejected.

Main-flow: The actor enters the business, picks the request, changes its status to "rejected" and explains the way, and then the request moves from the open requests list to the rejected requests list.

Alternative-flow: None

Positive test: After rejecting a request it will be moved to the rejected requests list and be removed from the open requests list.

Chapter 3 - Functional Requirements

1. In the system, at all times, there must be an admin.
2. An admin will be able to add a user (kibbutz member or nonkibbutz member as a business worker).
3. A kibbutz member will enter the system via a budget number and password.
4. Any kibbutz member will be able to send requests to a business.
5. Business workers will be able to reply to members' requests.
6. The admin will be able to track a kibbutz member's requests.
7. Each business will have a list of open requests, meaning requests that haven't been completed, and closed requests.
8. Business workers will be able to view all the details of open and closed requests of their business.
9. A member will be able to view the details of his open and closed requests.
10. There will be statuses to the requests.
11. There will be a possibility to restore a closed request.
12. There will be notifications: to the business' workers when a request has been received, to a member when his request has been read, and to changed status.
13. Each month a report will be sent from each business to the accountancy, including the expenses based on members' requests.
14. An admin will be able to add new businesses to the system.
15. A business worker will be able to manage his landing page, add photos, and manage an ad board.
16. A business will have its "cash register", that will document each expense made by kibbutz members in the business.
17. Accountancy will be able to view all businesses' incomes in a chosen time frame.

Chapter 4 - Non-Functional Requirements

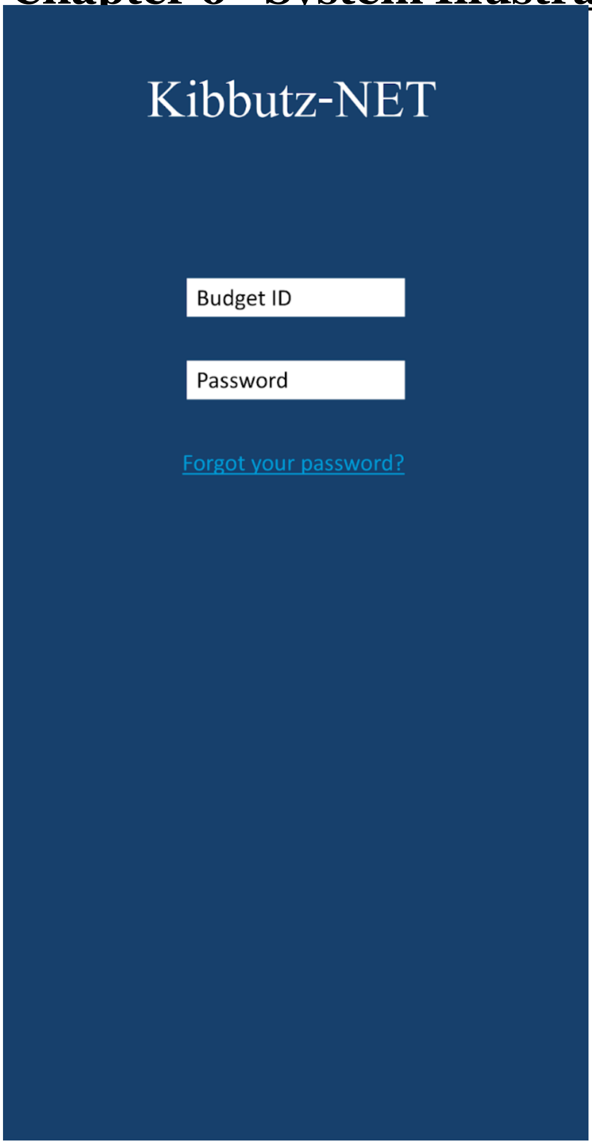
1. The system must be user-friendly, and suitable for the use of all kibbutz members. The use should be very simple and intuitive so that the system could be in use by real people who might have difficulties learning new systems.
2. The buttons and text will be large and clear, for older-generation users.
3. The report will be according to the accountancy’s requests.
4. The system should be suitable for computers, phones, and tablets.
5. The system should be able to handle the use of at least a few hundred users, with reasonable speed.
6. The database should be external and won’t be harmed when the server is disconnected.

Chapter 5 –
Risk assessment & Plan for the proof of concept

One of the major risks is the ability to integrate the system into pre-existing systems, which we are not going to implement as a part of our project, such as sending data to the accounting service our client is using now (SAP). Handling members' accounts are very sensitive, and there should be no mistakes. In addition, another major issue that may arise is the ability to make the system accessible to all users, especially to the older generation. The members should prefer to use our system instead of the current methods, which are outdated and have been discussed (with the client) and are not preferred and easy for both the members and the businesses.

The proof of concept will include a basic register, that can track orders and produce sheets, and also establish a conversation between two instances when one is a member, and the other is a business worker.

Chapter 6 - System Illustration

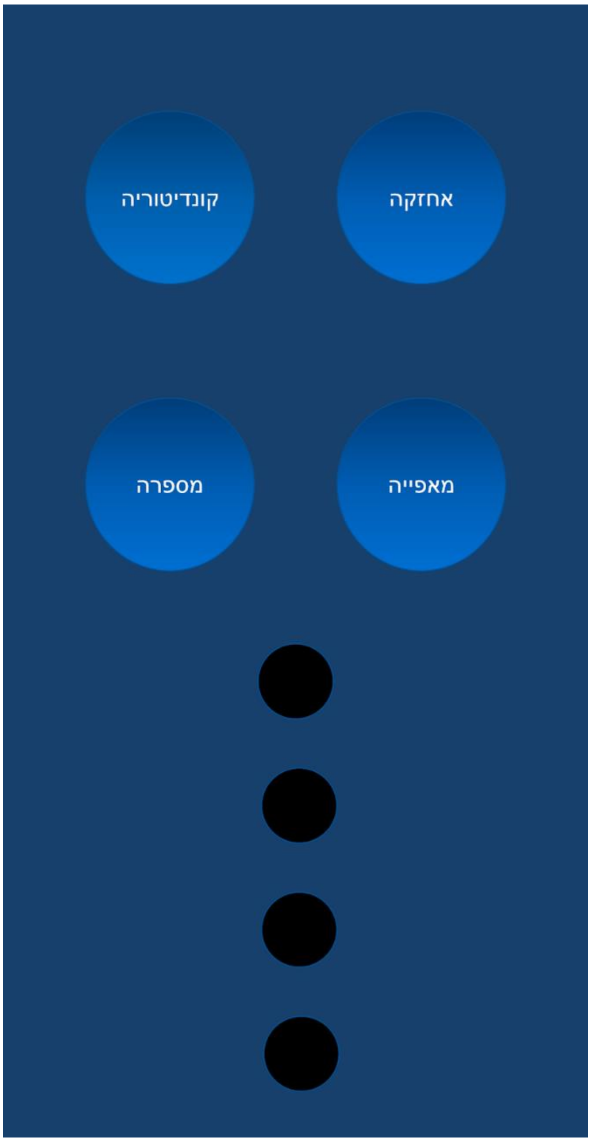


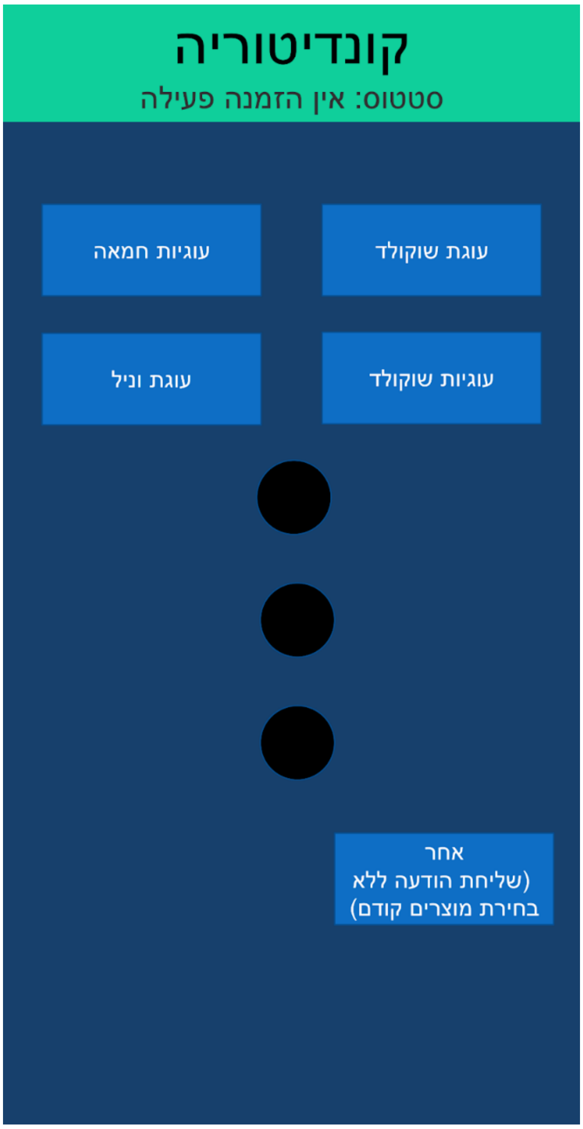
Kibbutz-NET

Budget ID

Password

Forgot your password?


אחזקה
קונדיטוריה
מאפייה
מספרה


קונדיטוריה

סטטוס: אין הזמנה פעילה

קונדיטוריה

- מבצע על עוגות...
- אנחנו סוגרים ביום שלישי הקרוב ה....
- עקב מחסור בחמאה העוגיות שוקולד.....

▪

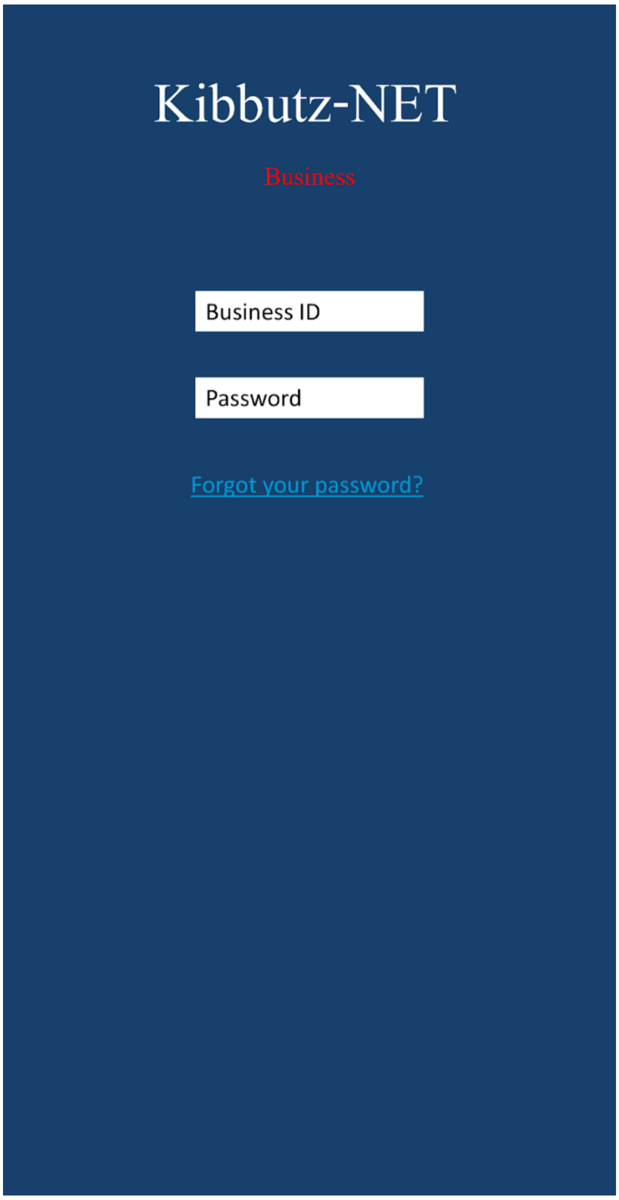
▪

▪

1

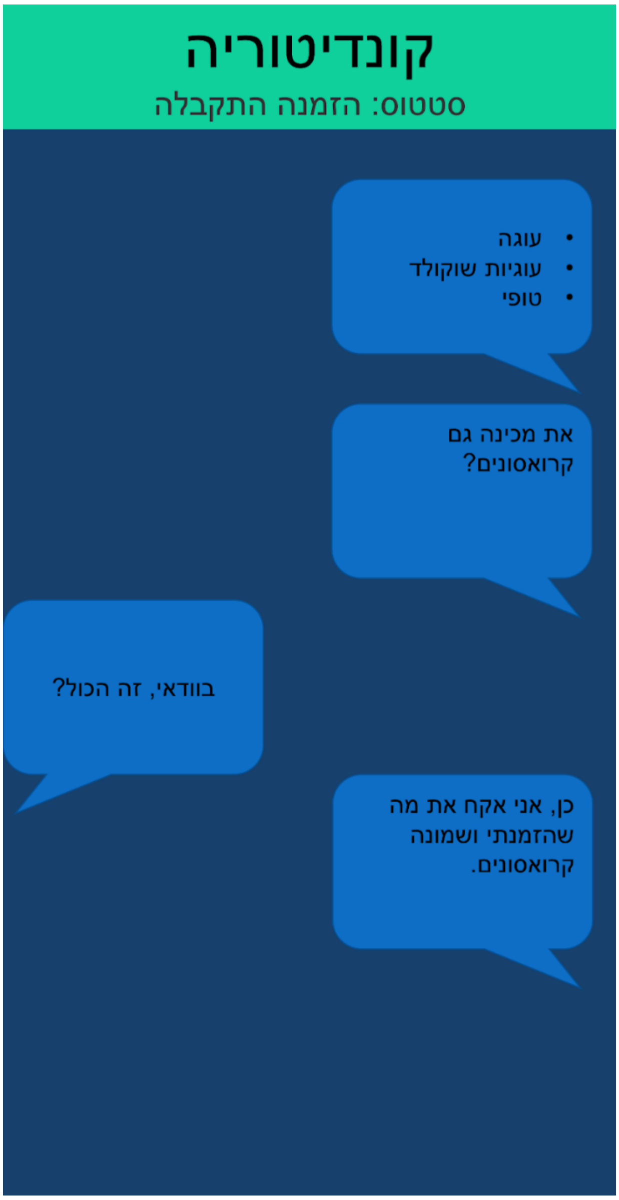
Chat &

Order



Kibbutz-NET

Business



קונדיטוריה

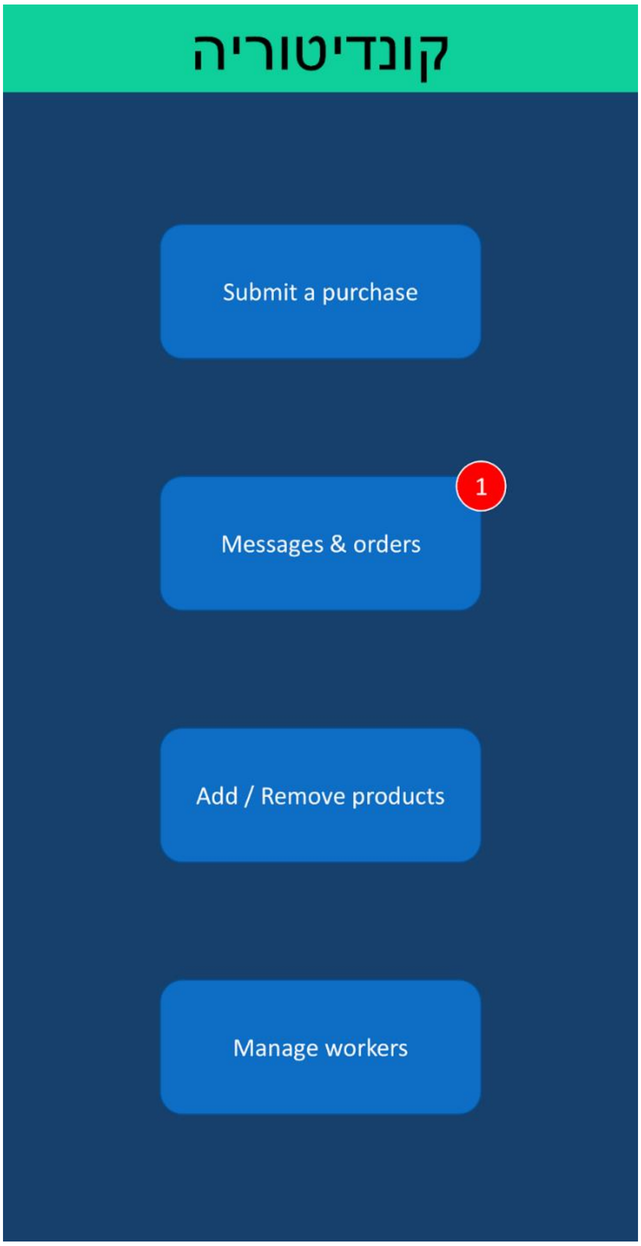
סטטוס: הזמנה התקבלה

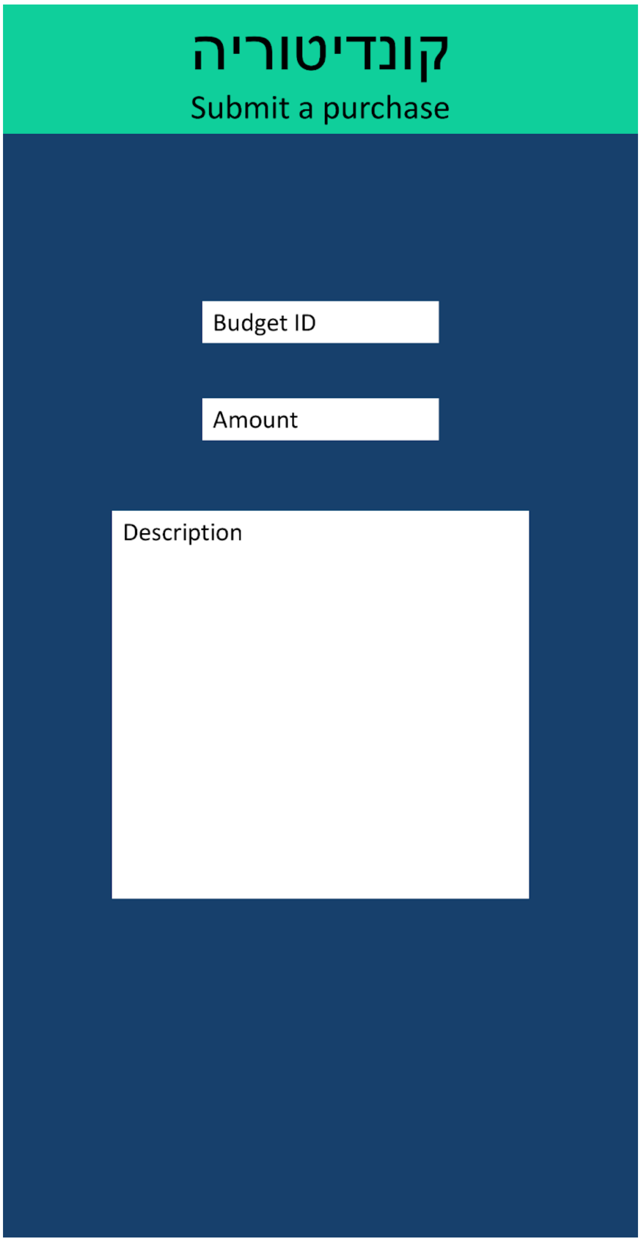
- עוגה
- עוגיות שוקולד
- טופי

את מכינה גם קרואסונים?

בוודאי, זה הכול?

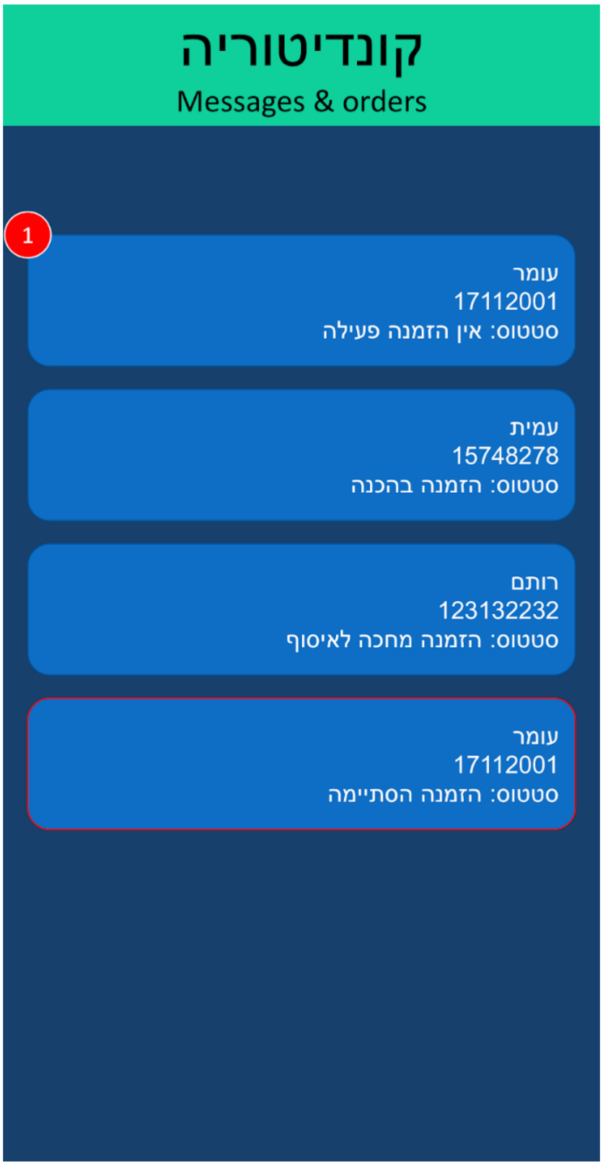
כן, אני אקח את מה שהזמנתי ושמונה קרואסונים.


קונדיטוריה
Submit a purchase
1
Messages & orders
Add / Remove products
Manage workers



קונדיטוריה

Submit a purchase



קונדיטוריה

Messages & orders

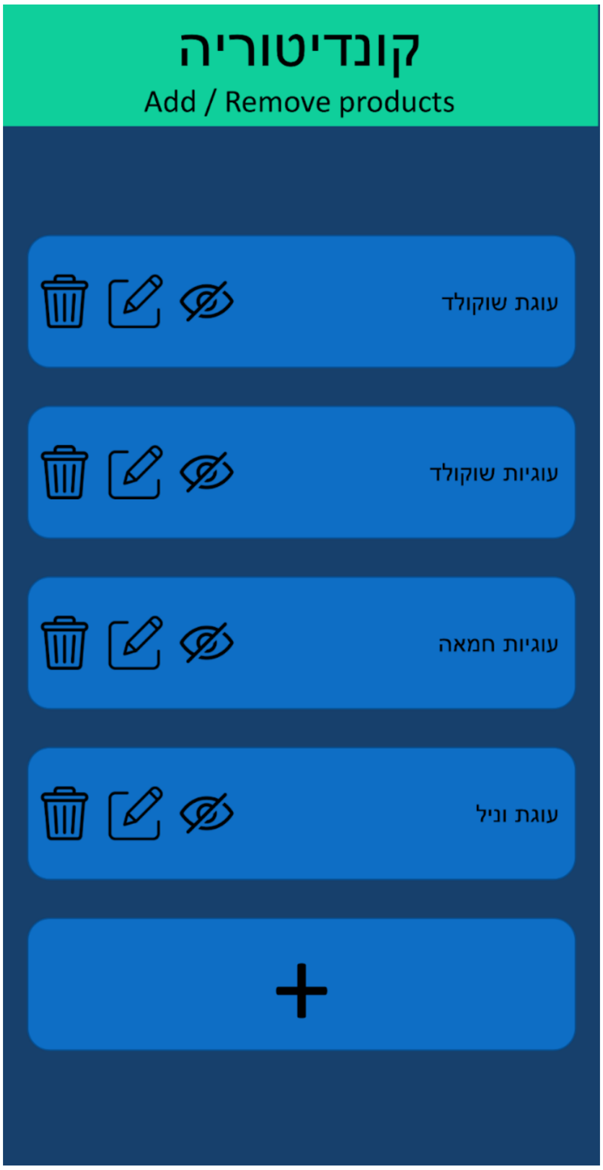
1

עומר
17112001
סטטוס: אין הזמנה פעילה

עמית
15748278
סטטוס: הזמנה בהכנה

רותם
123132232
סטטוס: הזמנה מחכה לאיסוף

עומר
17112001
סטטוס: הזמנה הסתיימה



קונדיטוריה

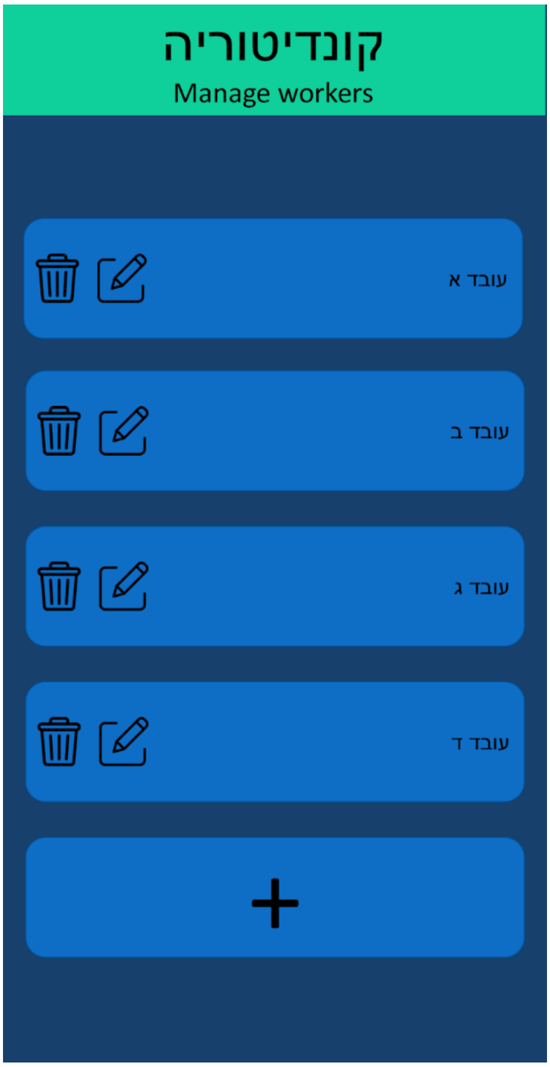
Add / Remove products

עוגת שוקולד

עוגיות שוקולד

עוגיות חמאה

עוגת וניל



קונדיטוריה

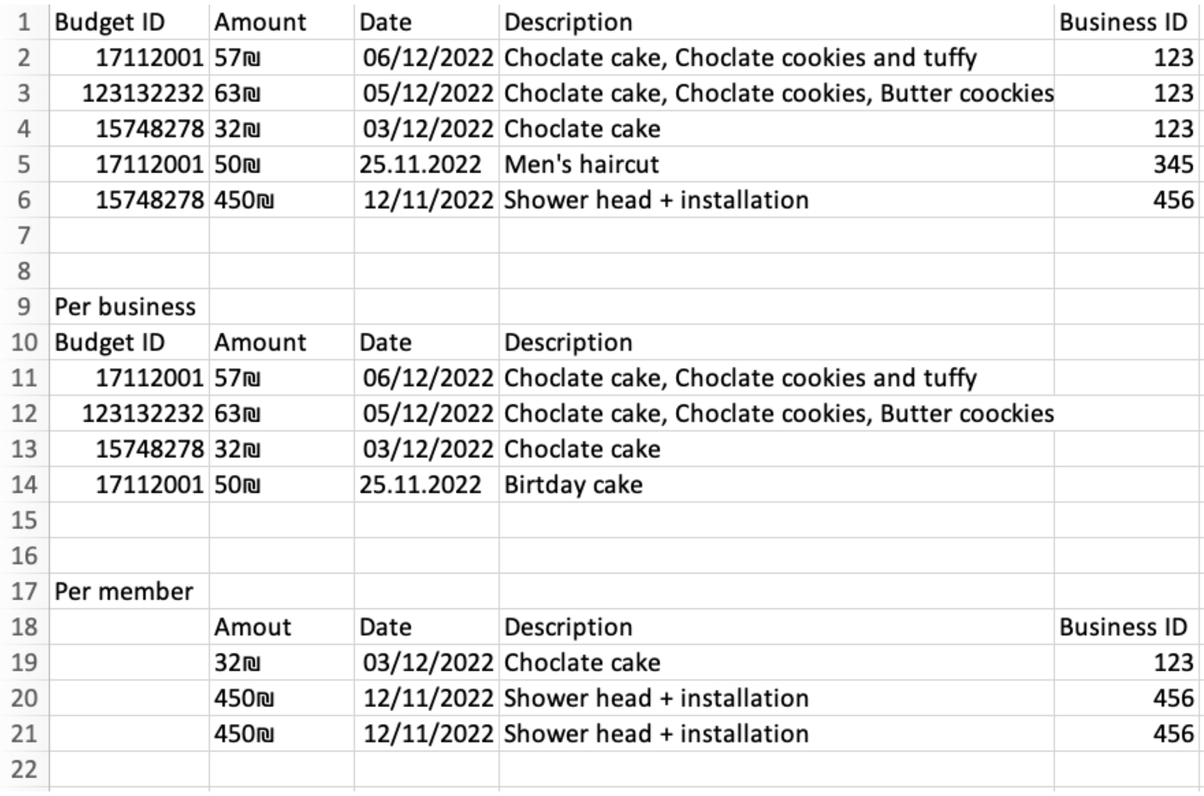
Manage workers

עובד א

עובד ב

עובד ג

עובד ד



Budget ID	Amount	Date	Description	Business ID
17112001	57₪	06/12/2022	Choclate cake, Choclate cookies and tuffy	123
123132232	63₪	05/12/2022	Choclate cake, Choclate cookies, Butter coockies	123
15748278	32₪	03/12/2022	Choclate cake	123
17112001	50₪	25.11.2022	Men's haircut	345
15748278	450₪	12/11/2022	Shower head + installation	456

Per business

Budget ID	Amount	Date	Description
17112001	57₪	06/12/2022	Choclate cake, Choclate cookies and tuffy
123132232	63₪	05/12/2022	Choclate cake, Choclate cookies, Butter coockies
15748278	32₪	03/12/2022	Choclate cake
17112001	50₪	25.11.2022	Birtday cake

Per member

	Amout	Date	Description	Business ID
	32₪	03/12/2022	Choclate cake	123
	450₪	12/11/2022	Shower head + installation	456
	450₪	12/11/2022	Shower head + installation	456