Date: 29-05-2016

Author: Kevin, Long, Petros, Eric

Version: 2.0

1 Introduction

1.1 Purpose of application

The purpose of this application is to create an online order system to reduce the waiting time of a queue. Customers are given the opportunity to look at an menu and order their food/drink etc through the app without the need of standing in a physical queue.

1.2 General characteristics of application

The program will be an online mobile application made for the Android platform. Internet is required to use the application.

The application will provide two graphical user interface. One for the customer and one for the Bar. The customer will be able to look at a menu and order their food and drinks through application. When the order is made the user will be able to see their queue number. There's also a chat where the user can talk with other users. If the user is a Bar, the user will be able to access the orders and push the virtual queue. The bar will also be able to edit the menu and create new Bars.

1.3 Scope of application

The application is limited to the users phones through the unique clientID. If no Bar is available the customer will not be able to enter the main application.

1.4 Objectives and success criteria of the project

- 1. The customer should be able to look through the menu and pick products to place an order.
- 2. The bar should be able to find the order and make it.
- 3. The bar should to be able to push the queue forward.

1.5 Definitions, acronyms and abbreviations

- Android - mobile operating system for smartphones
- Activity- class handling the view for every window in an Android application
- Bar - the user who handles the orders
- Customer - the user who makes an order
- ClientID - unique ID used to track the user of the application
- GUI - Graphical User Interface
- Product -the item a customer can order.

- Order- A list of products made by the customer.
- Queue- a queue based chronologically on the orders.

2 Requirements

2.1 Functional requirements

- Display a menu created by the Bar
- Customer can explore the menu options and the products
- Customer is able to place an order
- Customer gets an notification when it’s his or her turn.
- Customer is able to dismiss the order
- Global chat where customers can talk with each other.
- Bar is able to retrieve the orders
- Bar can confirm that the order is done and push the queue
- Authentication system to validate if it’s a Customer or Bar

Create a list of high level functions here (from the use cases).

2.2 Non-functional requirements

2.2.1 Usability

A lot of focus is put on the usability of the application. The application should be easy to learn and use for the customers. Especially drunk people who might have a hard time navigating through the application. The bar should also have an effective GUI to navigate through orders and change the menus.

User test will be used to make sure the application is easy to use. Design rules will be followed. E.g Schneiderman’s eight golden rules.

2.2.2 Reliability

The application shouldn’t crash. It should be hard for a customer to misplace an order. The bar shouldn’t be able to dismiss an order by mistake.

2.2.3 Performance

The application should be responsive and shouldn’t freeze. Although the performance might depend on the user's smartphone.

2.2.4 Supportability

The application will be implemented in a way to make sure most of the code is reusable and modifiable to suit other needs. E.g the GUI should be swappable from Android to Swing. The database will also be changeable.

2.2.5 Implementation

The implementation will be using the Java programming language which makes the code reusable on a lot of different platforms. The application itself will be made for Android smartphones. Every user will need to have the application downloaded.

2.2.6 Packaging and installation

The application will need to be downloaded in form of an APK.file and installed on the smartphone.

2.3 Application models

2.3.1 Use case model

See appendix for UML diagram

2.3.2 Use cases priority

- Add product to cart
- Make an order
- Update queue
- Authenticate user
- Receive order
- Edit menu

2.3.3 Domain model

See appendix for domain model.

2.3.4 User interface

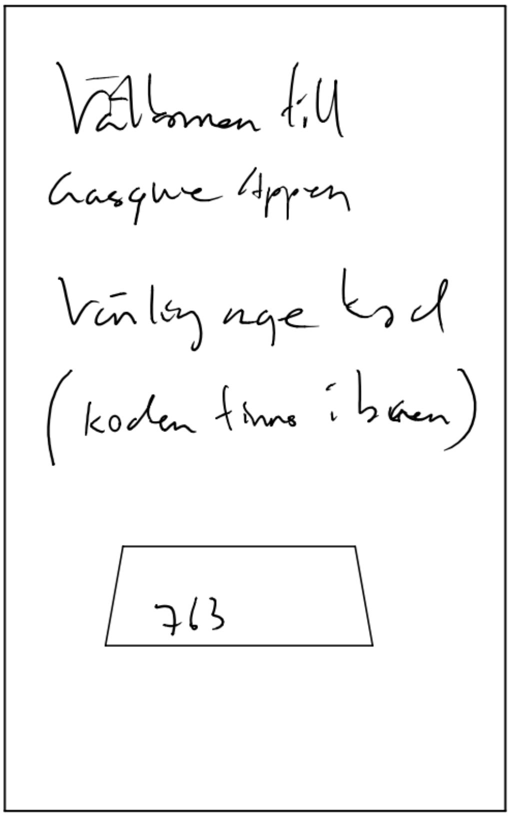
Clean UI, with grid of equals to showcase categories/items. Easy for the user to navigate by clicking on the grids.

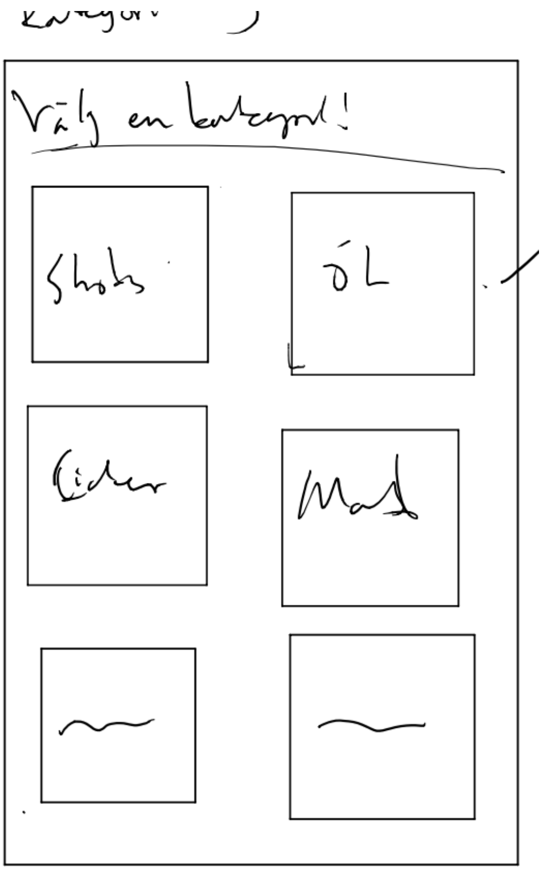
See appendix for ui design.

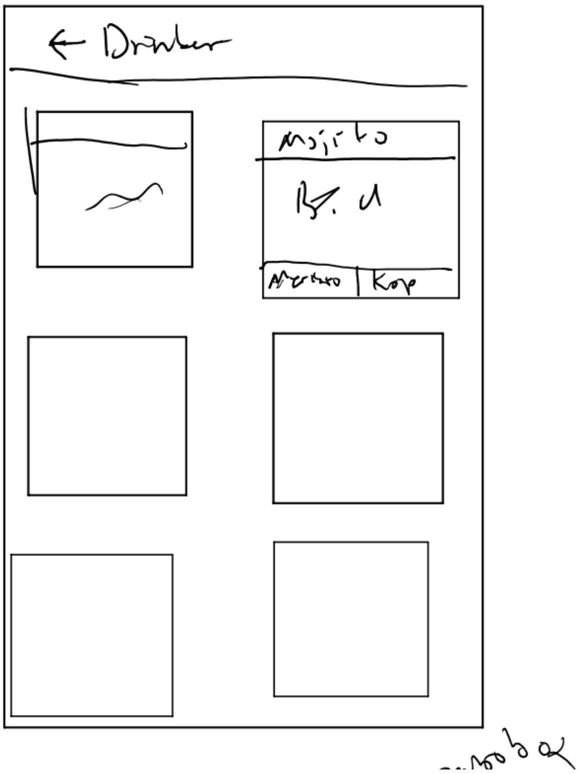
2.4 References

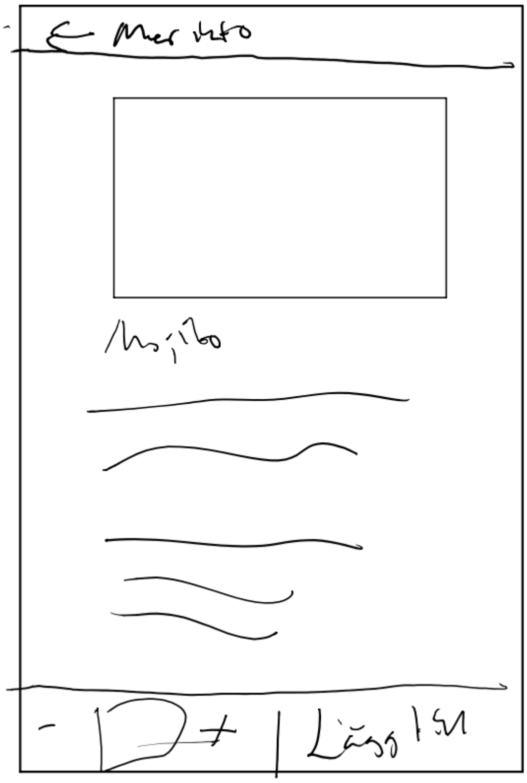
APPENDIX

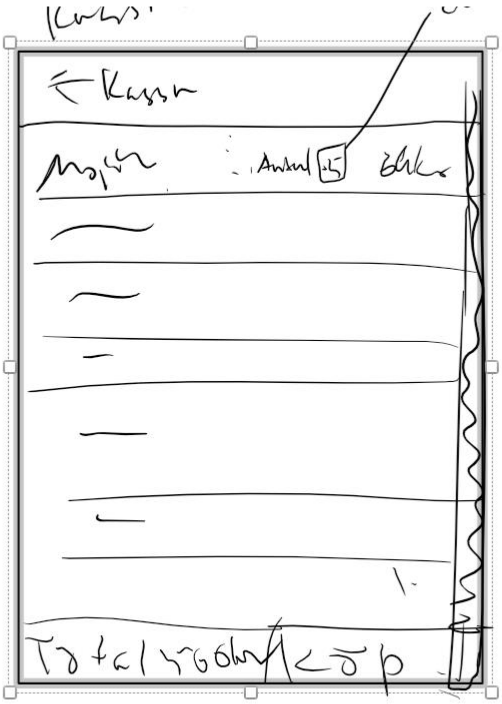
GUI

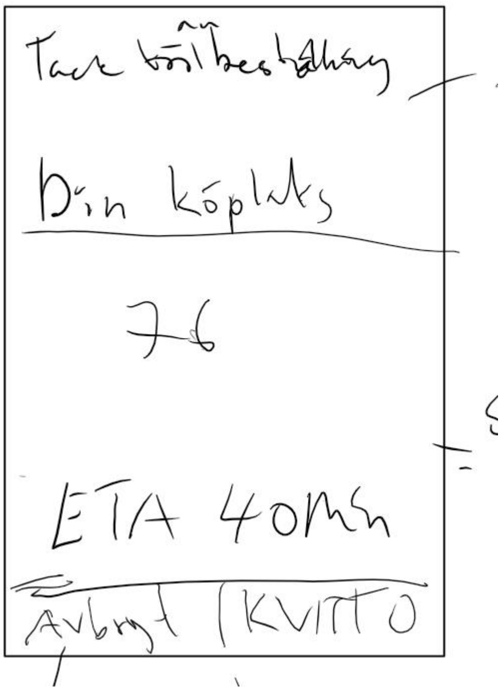

Välkommen till
Gasque Appen
Vänligen ange kod
(koden finns i baren)
763


kategori
Välj en kategori!
Shots
ÖL
Cider
Mat

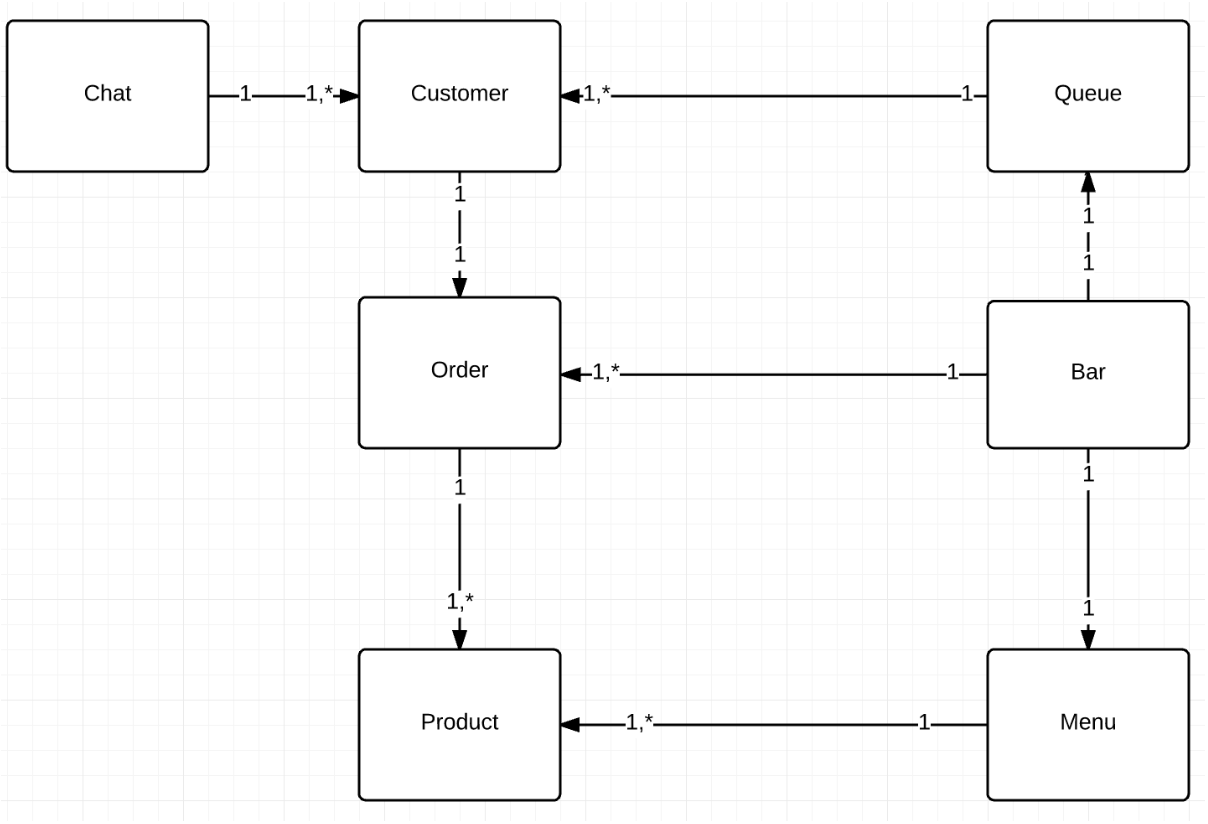

← Drinkar
Mojito
Bild
Mer info | Köp


← Mer info
Mojito
- 1 + | Lägg till

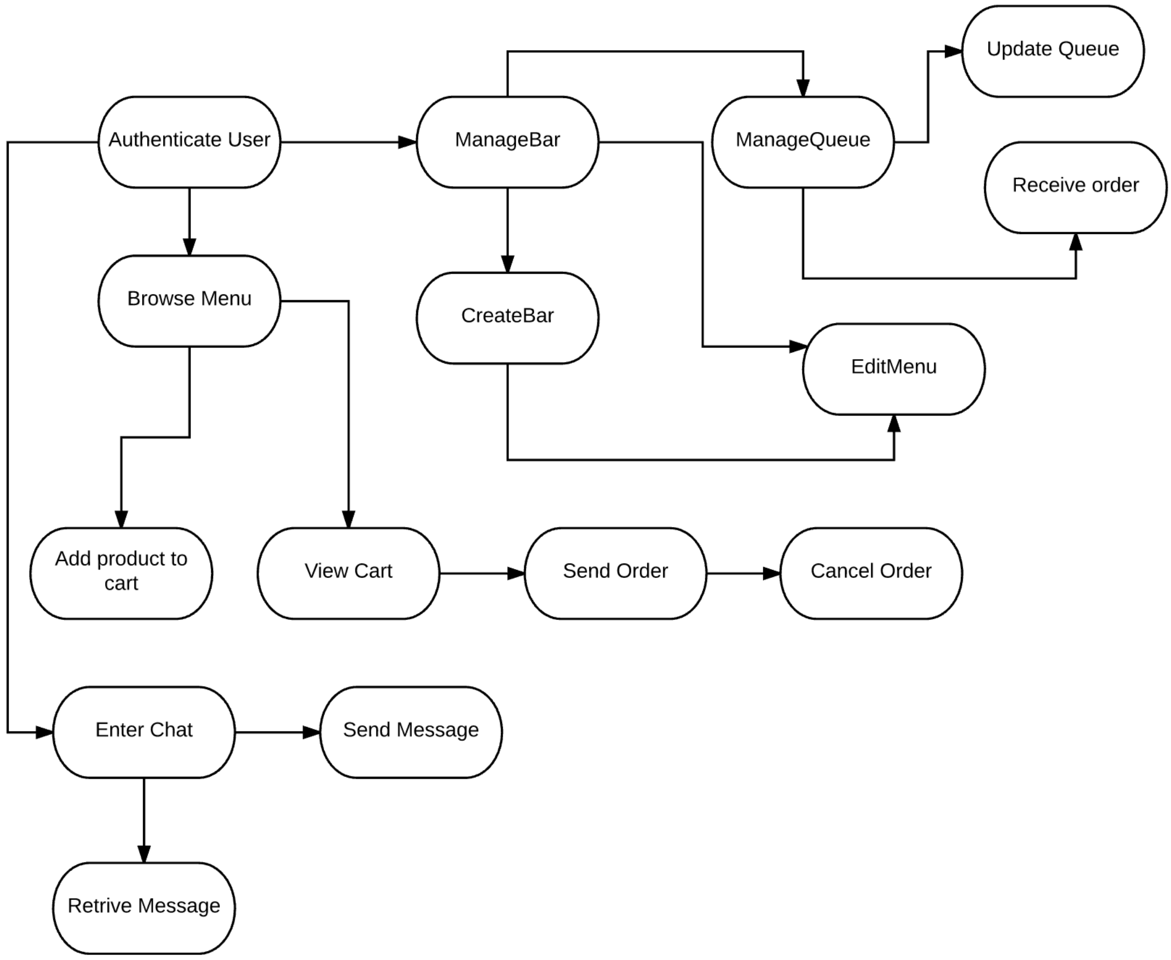




Domain model



Use Case UML-diagram



Priority Use cases listed below, other use cases - see separate use case document.

1. Add product to cart

Summary: Customer adds product to the cart that he/she intends to purchase.
Priority: High
Extends:
Includes: Browse menu
Participators: Customer

	Actor	System
1.		Displays product categories with grid of equals

2.	Clicks on one category	
3.		Updates view with a new grid of equals with products in chosen category
4.	clicks on product	
5.		Updates with view showing product detail
6.	Choose amount to buy	
7.	Clicks on add to cart	
8.		Cart updates with the added items

2. Make an order
Summary: Send cart of items to host
Priority: High
Extends:
Includes:
Participators: Customer

	Actor	System
1.	Clicks on cart button	
2.		Show items in cart view
3.	Clicks on confirm order	
4.		Send order to the host

3. Update Queue
Summary: Confirm that it’s a specific users time to pay
Priority: High
Extends:
Includes:
Participators: Bar

	Actor	System
1.	Confirm that bartender is ready for the next order by clicking the “push” brutton	

2.		Update list, send notification to the next costumer that is in line
3.		Update all users queue position.
4.		Countdown waiting time
5.	Confirms if payment is made or not	
5.1		If customer shows up, do nothing.
5.2		If customer does not show up, give customer a strike.

4. Authenticate user
Summary: User authentication deciding to give host or customer privileges in application
Priority: High
Extends:
Includes:
Participators: Customer, Bar

	Actor	
1.1	If Customer: Enter customer code	
1.2.	Clicks on verify	
1.2.1		If code is corret: Show customer main view
1.2.2		Else: Display code is invalid text
Alternate flow		
2.1	If host: Enter Host code	
2.2	Click on verify	
2.2.1		If code is corret: Show host main view
2.2.2		Else: Display code is invalid

		text

5. Receive order
Summary: Order saved locally to the host and is then displayed to him/her
Priority: High
Extends:
Includes:
Participators:Bar

	Actor	System
1		Order ends up on the database.
2		It is retrived from the database and saved to the bar’s app together with an id.
3		The order from firebase is cleared
4	Bar clicks the “push” button	
5		The next order in line is displayed on the bar’s phone

Alternate flow: List of orders is empty

	Actor	System
5		The old order remains on the screen

6. Edit menu

Summary: User is able to edit, remove and add products to menu.
Priority: High
Extends: managebar
Includes: createbar
Participators: Bar

Flow- edit product

	Actor	System
1.	Choose product category	
2.		Show product view
3.	choose product	
4.		show view to edit product
5.	edit product	
6.		saves edited product in menu
7.	clicks on create menu	
8.		updates database with new menu

Alternate flow: Creating new product

	Actor	System
3.	choose create new product	
4.		show view to create product
5.	click on create product	
6.		saves created product in menu

Alternate flow: Delete product

	Actor	System
5.	click on delete product	
6		product deleted from menu