

**Shenkar College of Engineering and Design**

**Faculty of Engineering**

**Department of Software Engineering**

**QBAR**

**Final project**

**By**

**Omer Goshen**

**Adam Galmor**

**Submitted as Part of the Requirements for receiving the**

**Bachelor of Science degree (B.Sc.)**

**{Date}**

Contents

[1. Introduction 1](#_Toc369019304)

[1.1. Overview 1](#_Toc369019305)

[1.1.1. Project Goal 1](#_Toc369019306)

[1.1.2. Problem 1](#_Toc369019307)

[1.1.3. Solution 1](#_Toc369019308)

[1.1.4. Future Development 1](#_Toc369019309)

[1.1.5. Audience 1](#_Toc369019310)

[1.1.6. Terminology 2](#_Toc369019311)

[2. Literature Survey 3](#_Toc369019312)

[2.1. Budweiser's "Buddy Cup" 3](#_Toc369019314)

[2.2. Guinness QR Cup 3](#_Toc369019315)

[2.3. Makr Shakr 3](#_Toc369019316)

[3. Architecture 4](#_Toc369019317)

[3.1. System users 4](#_Toc369019318)

[4. Design 5](#_Toc369019319)

[4.1. reacTIVision 5](#_Toc369019320)

[4.2. Perceptual 6](#_Toc369019321)

[4.3. Web 8](#_Toc369019322)

[5. Implementation 11](#_Toc369019323)

[5.1. Overview 11](#_Toc369019324)

[5.2. Details 13](#_Toc369019325)

[5.2.1. PerC implementation 13](#_Toc369019326)

[5.2.2. Web-server implementation 15](#_Toc369019327)

[6. Summary, conclusions and further work 16](#_Toc369019328)

[6.1. Summary 16](#_Toc369019329)

[6.2. Conclusions 16](#_Toc369019330)

[6.3. Further work 16](#_Toc369019331)

[6.4. System Weaknesses 17](#_Toc369019332)

[7. References 18](#_Toc369019333)

[8. Appendix 19](#_Toc369019334)

List of Figures

[Figure 1 - Budweiser’s “Buddy Cup” 3](file:///C:\Users\ogoshen1\Documents\QBAR_FPD.docx#_Toc368857338)

[Figure 2 – Guinness QR Cup 3](file:///C:\Users\ogoshen1\Documents\QBAR_FPD.docx#_Toc368857339)

[Figure 3 – Makr Shakr 3](file:///C:\Users\ogoshen1\Documents\QBAR_FPD.docx#_Toc368857340)

[Figure 4 –6 examples of reacTIVision fiducials 5](file:///C:\Users\ogoshen1\Documents\QBAR_FPD.docx#_Toc368857341)

[Figure 5 – Marking a bar section 6](#_Toc368857342)

[Figure 6 – Homepage screenshot 8](file:///C:\Users\ogoshen1\Documents\QBAR_FPD.docx#_Toc368857343)

[Figure 7 – Social page screenshot 8](file:///C:\Users\ogoshen1\Documents\QBAR_FPD.docx#_Toc368857344)

[Figure 8 – Menu screenshot 9](file:///C:\Users\ogoshen1\Documents\QBAR_FPD.docx#_Toc368857345)

[Figure 10 – Payment dialog screenshot 9](file:///C:\Users\ogoshen1\Documents\QBAR_FPD.docx#_Toc368857346)

[Figure 9 – Tab screenshot 9](file:///C:\Users\ogoshen1\Documents\QBAR_FPD.docx#_Toc368857347)

[Figure 11 – Architecture overview (from the Final Project Presentation in Shenkar) 12](#_Toc368857348)

[Figure 12 – Flowchart for the bartender use-case 14](#_Toc368857349)

# Introduction

## Overview

### Project Goal

The goal of this project is to provide a working implementation for a perceptual computing use-case; this prototype demonstrates how both a bartender and his customer can benefit from the use of technology.

It’s made up of several modules, allowing a customer to be identified and to place his order electronically, and it allows the bartender to serve to and bill his customer (at the same time), using gestures which are pretty much the same as his ordinary labor.

You might have seen robotic bartenders before, but this project is quite different, it aims to encourage human-human interaction, by making it more efficient and fun.

### Problem

Bartenders these days usually have to keep tabs on a computer. Those had touchscreens way before the majority of us had smartphones, and that makes sense. When you want things done in a bar or restaurant you can’t have the workers sitting around using the mouse and keyboard. Well today when most of us do have touchscreens in our pockets, they are still using the “same old thing”, they take our order, pour the drink, go and touch some prefabricated buttons on a touchscreen, and we get a little roll of paper with our tab on it.

And that’s the problem, by the time the bartender gets back, the customer can probably check his mail, update his Facebook status, twit some, and heck even play a round of Candy Crush…

### Solution

The answer is to take the bar to the next level again, and go touchless. That’s where Perceptual Computing technologies fit in, this project utilizes computer vision, facial recognition and hand gestures to help serve a drink faster than you can say Jack Daniels.

### Future Development

…

### Audience

…

### Terminology

* **Webapp** – the QBar mobile web application.
* **PerC –** Perceptual Computing. Usage of close-range hand gestures, finger articulation, speech recognition, face tracking, augmented reality experiences, and more, to make human interaction with computers more natural, intuitive and immersive.
* **Kinect** – is a motion sensing camera developed and manufactured by Microsoft for the Xbox 360 video game console and PCs. Based on a webcam-style add-on peripheral for the Xbox 360 console, it enables users to control and interact with it without the need of a keyboard, mouse, touchscreen or any outdate user control interface. This can be achieved through a Natural User Interface using gestures and spoken commands.
* **Gesture** - express an idea or emotion through bodily movements. It is a form of non-verbal communication in which visible bodily actions communicate particular messages, either in place of speech or together and in parallel with spoken words. Gestures include movement of the hands, face, or other parts of the body.
* **Q-Bar** – is a nickname for our smart perceptual bar.
* **Q-Bartender** – is a nickname for the bartender in the Q-Bar.
* **QR** – Quick Response code is a matrix code, 2-D bar code image, black and white Square that can be read by QR scanners and smartphones and mobiles phones with camera.
* **Fiducial** - In imaging technology, a fiducial marker or fiducial is an object used in the field of view of an imaging system which appears in the image produced, for use as a point of reference or a measure.

# Literature Survey



## Budweiser's "Buddy Cup"

This beer company has come up with a clever idea for another technological use for a beer glass.

Figure 1 - Budweiser’s “Buddy Cup”

These cups are equipped with a “bump sensor”, so when you tap your cup to another you and that person instantly become friends on Facebook.

It works by scanning a QR code on to bottom of the cup with a smartphone, which opens up Budweiser’s app, where you can link the cup to your profile.

## http://www.adweek.com/files/imagecache/node-blog/blogs/guinness-qr-code.jpgGuinness QR Cup

Figure 2 – Guinness QR Cup

The genius of this Guinness is that the QR code on the glass is literally activated by the beer.  
Pour a Carlsberg or Tuborg in there and you get nothing.

Scan the code with your smartphone, and it "tweets about your pint, updates your Facebook status, checks you in via Foursquare, downloads coupons and promotions, invites your friends to join you, and even launches exclusive Guinness content."

## http://images.gizmag.com/inline/makr-shakr-4.JPGMakr Shakr

Figure 3 – Makr Shakr

MIT’s Senseable City Lab is behind this robotic bartender, presented at Google’s I/O 2013 conference.

It’s actually much more than just that, users can mix their own drinks through an app on their smartphone and then share, learn and connect to others on social networks.

# Architecture

## System users

There are only two types of users in the system, a bartender and his customers.

#### Bartender

This user is the only one tracked by the Kinect, his inputs to the system are his hands and coasters which he puts on the bar. He has a monitor and speakers to get feedback from the system.

#### Customer

Users of this type use the mobile web application on their smartphones for most of their actions, but can also wave their hand and show their face to the “hostess” (virtual, it’s a camera and face recognition module).

# Design

The system contains several distributed modules, communicating over the network.   
The design is mostly the result of constraints (in hardware & software, explained in depth later on), but also follows best practices and methods of software development.

### reacTIVision

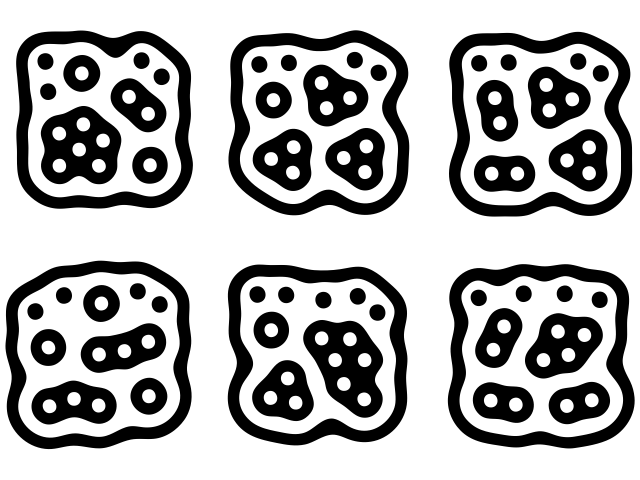
This is a proprietary piece of software that uses computer vision to identify special markers (*Fiducials,* Figure 4). It runs as a standalone application which sends [Open Sound Control](http://opensoundcontrol.org/introduction-osc) messages over pre-configured UDP ports. These messages contain the ID of the fiducial, its 2D position and orientation, and a timestamp.

Figure 4 –6 examples of reacTIVision fiducials

reacTIVision actually is the “brain” which operates the reacTable, an audio synthesizer with touch UI.

Fiducials were chosen over QR codes for their fast recognition and increased camera range.   
The poorly lit, something smoky environment of a bar makes computer-vision harder and also disqualifies solutions that use color. Fiducials are based on contrast and hence are color agnostic.

### Perceptual

This module uses a Microsoft Kinect camera to track the bartender’s hands and recognize his gestures. It’s was written in Java\Processing using the OpenNI library.  
This module also serves as a client for the reacTIVision server, so only the combination of the bartender’s hand inside a bar section and a message from reacTIVision the coaster on the bar triggers the completion of a customer’s order.

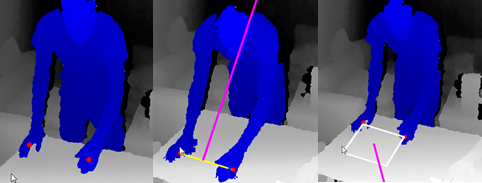


Figure 5 – Marking a bar section

The above images (Figure 5 – Marking a bar section) are screenshots of the running Processing sketch, they show a sequence of depth images from Kinect, with the tracked user colored in blue (courtesy of the OpenNI library). The user is performing a setup for a bar section, it’s consisted of marking the corners of a quad with his two hands.

Several gestures for performing this action were theoretically considered, neither were practical for implementation for various reasons.  
Using machine-learning algorithms with a large training set is above the scope, expertise and budget for this project.

A possible method could be “dwelling” in the desired locations, while not nearly as hard to implement as the previously suggested method, it adds a lot of time to testing and usage.

For the results of marking a bar section to be accurate, meaning that is aligns to the plane of the bar, some extra work is needed. Once again, this is beyond the scope of this project, but a suitable method can be found in parts of various “whittling” solutions, those applications that reconstruct objects or environments when filmed by a depth-sensing camera such as the Kinect. It usually revolves around usage of the RANSAC algorithm to detect planes inside the point cloud.

As stated before, this project’s scope doesn’t span the features needed to make the gesture natural, intuitive nor immersive. The workaround was to use a mouse or keyboard, pressing the right mouse button (which when is a wireless mouse, a single user can hold in his hand while performing the gesture) or the spacebar on the keyboard (requires a second user).

There’s also support for deleting a section, for when the process results in a “bad” quad, which doesn’t align with the surface. The user can left-click a section with the mouse and press ‘delete’ to remove it.

### Web

Written in Java\Processing, this module uses the [Jetty](http://www.eclipse.org/jetty/) web-server and [Cometd](http://cometd.org/documentation/cometd-java) server-push library to host the mobile webapp. On the client-side, a responsive HTML5 webapp allows the customer to order drinks and pay his tab.

When the client reaches his seat he can decide his preferred way of placing an order. All he needs to do in order to use the webapp for it is to wave his hand from side to side. This will generate and display a unique QR on the screen in front of him, once the QR is scanned by a mobile device the webapp initializes with his unique token. You can see this process as a modern hostess in contrary to today’s busy human hostess.

#### User Interface

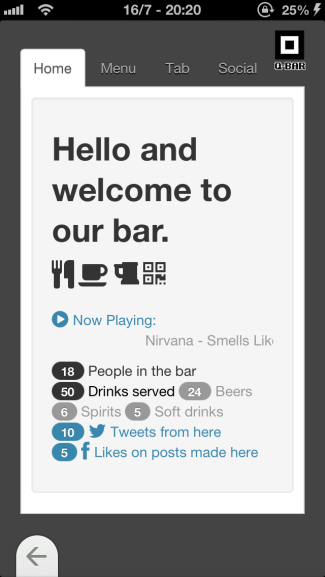


Figure 6 – Homepage screenshot

The webapp contains several tabs the first is the “Home” tab (Figure 6 – Homepage screenshot), its goal is to expose the user to the main social activities occurring at the local Q-Bar. It does that by displaying the currently playing song, the number of tweets, posts and check-ins made at the bar.   
For a worldwide point of view, the user can preview for example during a football game, the number of beers served at his favorite team homeland Q-Bar.

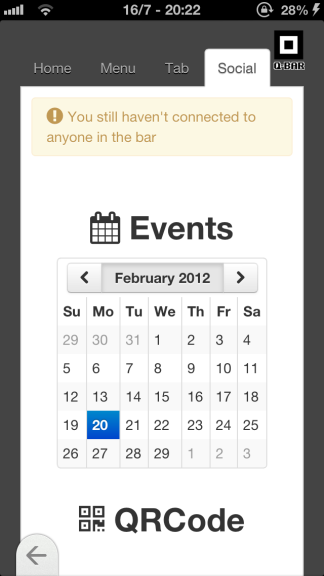


Figure 7 – Social page screenshot

Another social related tab is the last one named “Social”; its purpose is to display the users’ social networks events on a calendar, merged with highlighted Q-Bar events.

Below the calendar a QR is displayed, this is the same QR the user got upon his first interaction with the webapp. Its purpose is to enable a bill sharing system, if for example the client arrived with a friend to the Q-Bar and that friend decided he wants to pay the bill, he can then scan that QR and the bill tied to his friend’s token will be displayed on his mobile ready to be paid!

The second tab is the “Menu” (Figure 8 – Menu screenshot); it displays a list of beverages, their cost and an editable quantity input box. When the client presses the order button at the bottom of the page, the order is placed.   
The client is charged only when the Q-Bartender displays the right QR coaster to the image recognition camera.

This way the client can be pinged on delivery (Figure 9 – Tab screenshot).

Figure 8 – Menu screenshot

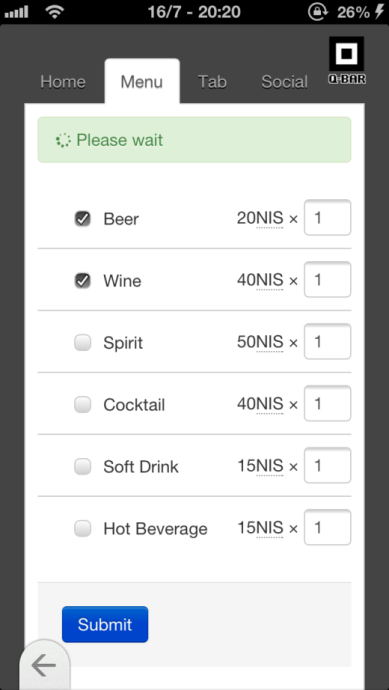
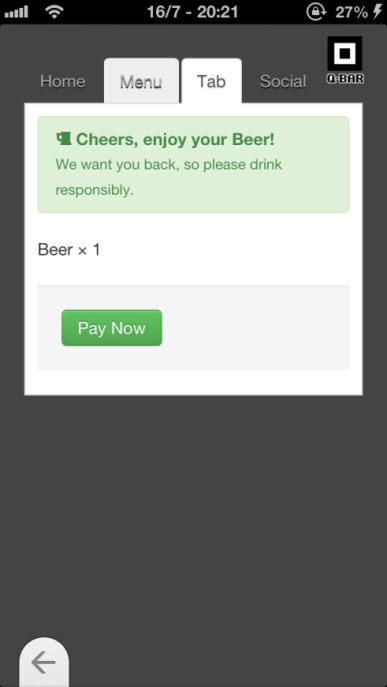
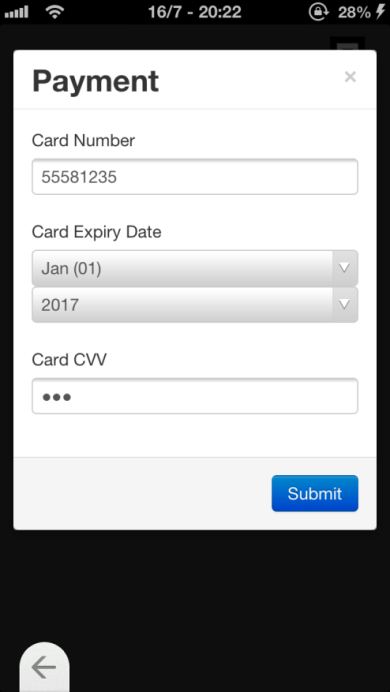


Figure 10 – Payment dialog screenshot

Figure 9 – Tab screenshot

Upon charge the next tab “Tab” (checkout) gets focused describing the bill and enabling a payment option. The client can pay by entering his credit card credentials.

# Implementation

## Overview

An overview of the system is presented below (Figure 11); it shows the various components of the system with an illustration of their network topography.

Starting from the top of the diagram, on the left of the cloud you can see the various HTML pages that make up the QBar webapp, the arrows connecting them to the customers’ smartphones are bi-directional, which is an insinuation for the “Server-Push” technology used, it’s backed up the Java web-server (CometD) and uses a corresponding jQuery plugin.

Phones, representing unique users are actually distinguished through different codes; the QR codes are also bi-directionally connected to suggest the social aspect of the system.

On the bottom of the diagram, from left to right, we see the reacTIVision server connected to a regular webcam; it sends messages to a preconfigured host and port over UDP (which camera and ports are configured in camera.xml and reacTIVision.xml respectively, see the reacTIVision documentation for more information).

The perceptual computer is connected to depth-sensing cameras, which are continually queried for the users’ hands positions, gestures (such as waving a hand) and identity (in the form of facial recognition, whose usage wasn’t implemented in the scope of this project).

Finally, the web-server, hosts the various markup, stylesheet and script documents that make up the QBar webapp. It serves resources in a RESTful manner to AJAX requests made from the webapp clients over HTTP, again, using jQuery on the client.

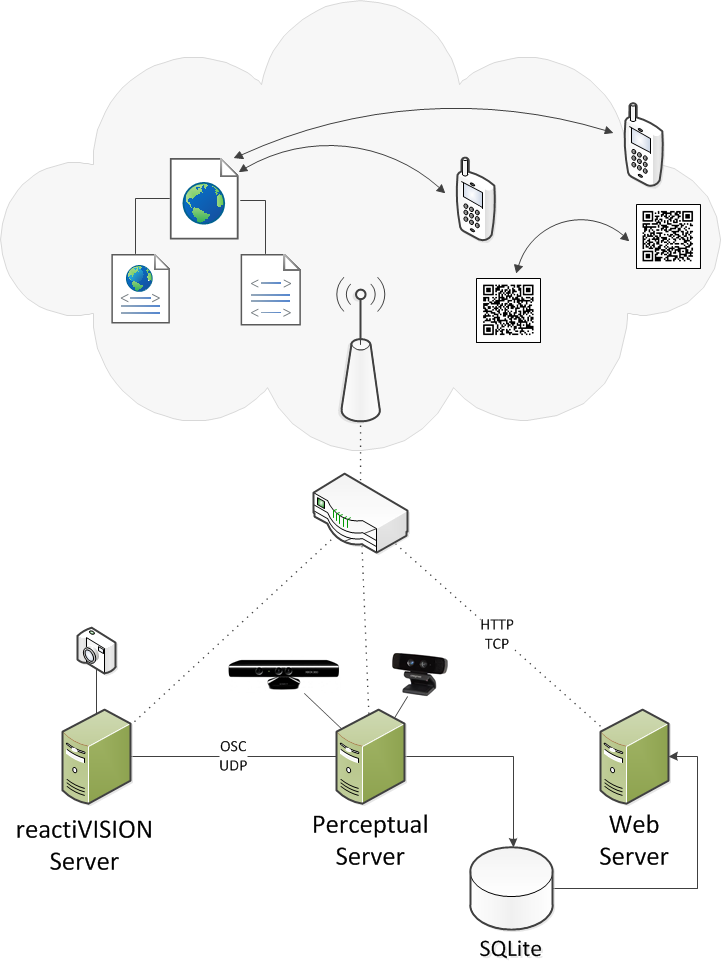


Figure 11 – Architecture overview (from the Final Project Presentation in Shenkar)

## Details

### PerC implementation

The main flow of system is illustrated below (Figure 12, Taken from the FRS document).   
It describes how the drinks served by the barman are tabbed.

Starting conditions are that the user is recognized by the Kinect and that bar sections are defined.

The system then goes into a loop, sampling the user’s hands and checking whether they are inside a bar section (elaborated later on).

When that check returns true, we can proceed to query the reacTIVision server for the presence of a coaster in the active section.

When both conditions are met, the drink will be tabbed to the current section.

As described in the Perceptual chapter of the Design section (4.2), bar sections are quads in the Kinect 3D world-space, to find the active one we calculate the Euclidean distance from the center of each section to the user’s hand. We pick the closest section, but only mark it as active when the distance is below a constant value.

#### Installation notes

To actually use the Kinect on a Windows PC, the OpenNI SDK has to be installed, along with the NITE middleware.

The SimpleOpenNI library wraps these for use with Processing (newer versions depend on the Microsoft Kinect SDK instead).

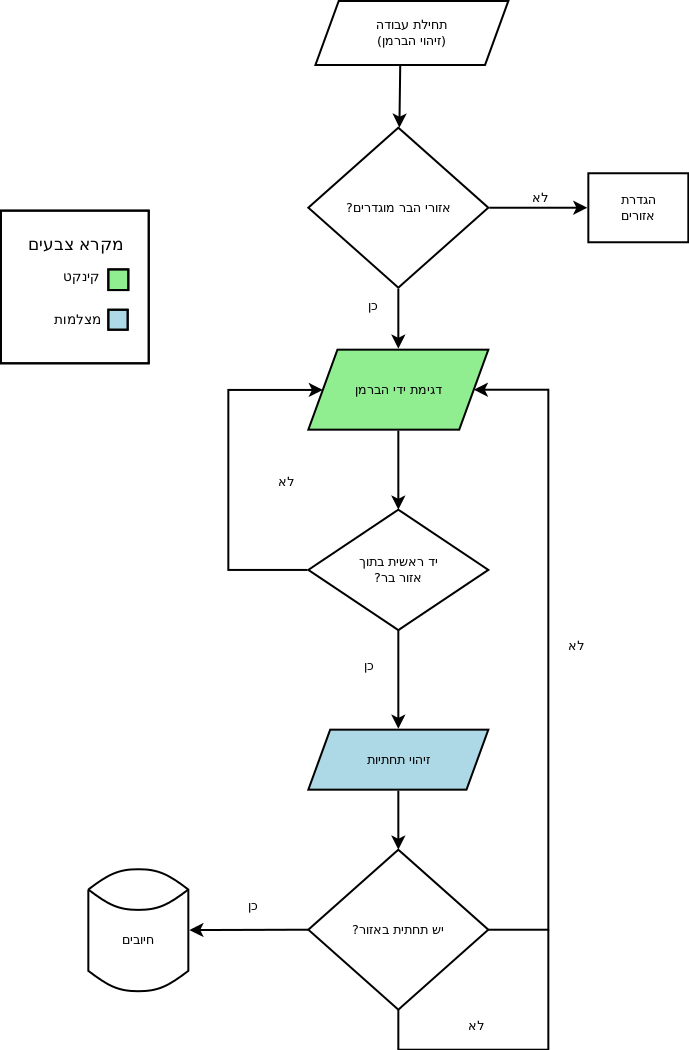


Figure 12 – Flowchart for the bartender use-case

### Web-server Implementation

Jetty is a lightweight web server implementation in Java.

The various aspects of the server were coded in classes which handle the HTTP requests, the following is a description of the Handler interface implemented by these classes (From the Jetty API docs):

A [Handler](http://download.eclipse.org/jetty/stable-9/apidocs/org/eclipse/jetty/server/Handler.html) instance is required by a [Server](http://download.eclipse.org/jetty/stable-9/apidocs/org/eclipse/jetty/server/Server.html) to handle incoming HTTP requests. A Handler may:

* Completely generate the HTTP Response
* Examine/modify the request and call another Handler
* Pass the request to one or more other Handlers

Two handlers were written to deal with requests, the first responds to the ‘qr’ query parameter with a 250x250px PNG image of a QR code.  
The second returns a JSON representation of an order as sent by the webapp.

CometD is implemented as a servlet. Each request by the client spawns a thread on the server, which spins in a loop.  
The body of the loop checks whether the message from reacTIVision was received and if so, it echoes a JSON representing the drink and breaks.

The internals of the CometD servlet takes care of keeping the HTTP connection open, and only sends the headers of the response when all processing of the request is over (when the loop ends), this is how “server-push” is implemented.

### Webapp Implementation

Bootstrap

jQuery

Backbone

# Summary, conclusions and further work

## Summary

Increase the Q-Bartender’s client socializing aspect.

As a result of growing demand from the clients’ side for a faster and better service and on the other hand the Q-Bartender’s multitasking difficulties we’ve decided to relive him from nowadays’ touchscreens shackles and march him straight to the 21th century practical UI through the use of perception “gimmicks” and the social web application.

## Conclusions

We managed to achieve a more intuitive and immersive way of drinking and serving the client’s demands. The bartender no longer haves the need to go back and forth to touch screen placing his orders or generating the bill, to do so he can simply use perceptual gestures. As a result we belief the user experience of both the bartender and the client will increase.

In conclusion the bartender is free again to socialize with the customer making him feel interested. The client is aware of the surrounding social events and can choose wither or not he is willing to participate.

## Further work

* **Social rating of orders in the bar**To increase the client’s social immersion it is possible to add the ability to view what others have ordered, their drinking habits and drinks rating. In general the idea is to create for every customer a Q-Bar profile so that customers could see what others have ordered and how much they liked it.
* **All-you-can-drink feature with a special QR'ed glass bottom.**If we have an image recognition camera recording the customer nonstop, why not use it to watch the cup as-well. We can print the relevant drink’s QR at the bottom of the cup, once the cup is empty the QR is revealed and a ping is sent to the billing system, telling the Q-Bartender to refill the client’s cup.   
  This feature can be toggled by the web application or verbally through the Q-Bartender.
* **Network of Q-Bars**Have drinking contests with other Q-Bars worldwide, while watching a sporting event
* **Gamification feature**Grant users with experience point through the use of the system so they can latter on trade them for discounts and Q-Bar private events. This will encourage the regular customers to participate and break their technological phobias.   
  Receive experience points by using the web application to.
* **Image recognition camera theft security.**The stolen smartphone market is growing day by day, making people les free at the bar to socialize. It is possible to use a face recognition algorithm to attach the client’s smartphone to him. Once the user leaves his seat and the smartphone is no longer present inside the virtual customer area, an alarm will be raised a photo of the thief is captured so the Q-Bar security personal can trace and deal with the incident.
* **Voice recognition**The Kinect camera is installed with four microphones to reduce the surrounding noise and be able to perform voice command. This feature can be harnessed both for the client’s and Q-Bartender’s voice commands.

## System Weaknesses

* **The Kinect camera is limited to a seven meters vision.**Another vision disadvantage at the bar is the fuggy and dark environment, making it tough to recognize the drink QR coaster and perception gestures.
* Finding new and original hand gestures for the Kinect camera is not intuitive as it may seem.
* **The system as a whole is difficult to set up**Cameras of different nature should be placed in strategic points at the bar which can sometimes be missing or impossible to find.
* **Customers’ technophobic nature**Might repel the use of recording cameras pointing at them the whole time, capturing and documenting their every move. The social aspect can be intimidating, check-ins at the bar are public and can be seen by unwanted eyes.

# References

# Appendix

תקציר ודוגמאות בעברית

**רעיון הפרויקט**

נולד כחלק מהשתתפותנו בקורס המשולב עם מחלקת תקשורת חזותית בשנקר. חיפשנו מקוריות ופריצת דרך בתחום החישה הממוחשבת ועולם המציאות הרבודה. בהתחלה חשבנו על רעיונות כגון "ארון בגדים ווירטואלי" בו המשתמש רואה על גבי מסך תמונת דמות אווטאר בה הוא לבוש באותם הבגדים שבחר. רעיון נוסף היה "שולחן עבודה במרחב", רצינו לשבור את המוסכמות של ימינו בהם ריבוי מסכים הינו הפתרון הטריוויאלי להגדלת שטח שולחן העבודה. שאפנו את ההשראה לגבי רעיון זה מסרטים כגון "דו"ח מיוחד" בו השחקן הראשי משתמש בכפפה כדי להזיז עצמים המופיעים על המסך מולו. הסרט "איש הברזל" לקח רעיון זה צעד אחד קדימה, על ידי השימוש בהולוגרמות עצמים ומיקומם במרחב. אך כל אלה ועוד אינם רעיונות מקוריים ולכן החלטנו לעסוק דווקא ברעיון הבאר החכם.

**הבעיה**

נסו לחשוב איך התנהל עולם המסעדנות לפני כניסתם של מסכי המגע. הוספת הזמנה חדשה במערכת דרשה מהברמן/מלצר להכניס פריט אחר פריט בעזרת המקלדת. לאחר מכן, נוצרו מערכות חיוב הכללו את שלל הפריטים הקיימים בבר ובעזרת שימוש בעכבר וסימון הפריטים נוצר החשבון, אך לדעתנו זו "אותה הגברת בשינוי אדרת". אנו רוצים לחסוך מעובדי הבר את הצורך ללכת הלוך ושוב למסוף הקופה בכדי ליצור, לעדכן ולסגור את חשבון הלקוח. בעזרת השימוש במחוות ומצלמות חישה ניתן להוציא מן המשוואה את קשיי העבודה הללו ולהפנות את אותם המשאבים והזמן ליצירת אינטימיות ותחושת חברותא בין הברמן ללקוח. רצף האירועים ללקיחת הזמנה אינו נקטע כך שהלקוח מקבל תחושת אירוח מלאה יותר מהברמן.

**המטרות**

המטרות אותן שמנו לפנינו היו יצירת זמן חברותא עבור הברמן על ידי ביטול הצורך שלו להתעסק עם הקופה. השגנו מטרה זו בעזרת השימוש במצלמות החישה וזיהוי תמונה בבאר, כך שהוספת פריט לחשבון הלקוח נעשית בעזרת חשיפת תחתית המציגה את הברקוד המתאים למצלמת הרשת ובעזרת אלגוריתם לעיבוד תמונה מתווסף הפריט לחשבון. הנפקת החשבון עבור הלקוח נעשית בעזרת מחוות ניפוף וביד השניה סימון העמדה הרלוונטית. באפשרות הברמן ליצור עמדת “Happy Hour” וכאשר ידו נמצאת בתוכה, מחירי המשקאות בבר מוזלים. את העמדות הווירטואליות יוצר הברמן בתחילת המשמרת כאשר הוא מזדהה בפני מצלמת זיהוי התנועה (Kinect) בעזרת מחוות ניפוף שתי ידיו, לאחר מכן ידיו מסמנות במרחב 4 נקודות המגדירות את עמדת הלקוח. מעל עמדה זו אנו מציבים מצלמת רשת המזהה את תחתית המשקה בעלת הברקוד היחודי ושולחת אותו למחשב המרוחק לעיבוד הזיהוי והוספת הפריט לחשבון.

במהלך הפרוייקט רצינו ליצור פלטפורמה מלאה עבור הלקוח, פה נכנסת לתמונה האפליקציה הרשתית. כאשר מגיע הלקוח לבר קיימת עבורו האפשרות לשימוש באפליקציה זו ובכך לחסוך, במידת הצורך, את התור בכניסה למקום ואפילו את התור להזמנת המשקה. בכניסה לבר ממוקמת מצלמת זיהוי תמונה נוספת המנפיקה ללקוח בעזרת מחוות הניפוף, ברקוד יחודי. כאשר סורק הלקוח עם מכשיר הטלפון החכם שלו את אותו הברקוד נפתחת במכשירו אפליקצית הרשת המשוייכת לחשבונו הפרטי, בעזרתה יכול הוא ליצור הזמנה חדשה, לראות את רשימת האירועים בבר על גבי לוח שנה, רשימת השירים המתנגנת בזמן אמת בבר ונתונים מרשתות חברתיות, כגון טוויטים ופוסטים הנעשים במקום.

**דוגמאות לפרוייקטים**

ישנם שיגידו כי פרוייקט זה מקדים את זמנו ואינו פרקטי בימינו אך בפרספקטיבה בין לאומית אנו רואים יותר ויותר חברות השמות לעצמן כמטרה להוביל את התחום החברתי והשירותי בענף המסעדנות וחיי הלילה. הצגנו במהלך המצגת וספר הפרוייקט מספר דוגמאות:

חברת Budweiser השיקה לא מזמן את ה"כוס החברתית" שלה הנקראת “Buddy Cup” היוצרת פלטפורמה ל"שבירת הקרח" בבר. כל שנדרש מהלקוח הוא לסורק את הברקוד בתחתית הכוס ובכך לשייך את חשבון הפייסבוק שלו אליה. בנוסף בתחתית הכוס מושתל רכיב לשידור אלחוטי (Wi-Fi) ומד תאוצה (Accelerometer), כאשר שתי כוסות פוגעות אחד בשניה הופכים הלקוחות לחברים בפייסבוק.

חברת Guinness גם כן מנסה לתפוס את מקומה בעולם החדשנות של ימינו, היא עושה זאת בעזרת ה “Guinness QR Cup”. כוס עליה מודפס ברקוד יחודי אותו ניתן לסרוק רק כאשר הכוס מלאה בבירה היחודית המקנה לו רקע שחור המסמל את בירת המותג. בפעולה זו נשלחת הודעה לחברי הלקוח על נוכחותו בבר, הזמנת חבריו, קבלת הטבות ועוד.

באוניברסיטת MIT פיתחו רובוט בשם Makr Shakr הממלא את מקום הברמן. לקוחות יכולים ליצור את מתכוני הקוקטלים המעודפים עליהם בעזרת אפליקציה יעודית בסמארטפון שלהם עבור הרובוט ולאחר מכן לשתפם ברשתות החברתיות.

חשוב להדגיש כי שמנו לעצמנו מטרות אקדמאיות לההוכחת הרעיון (proof of concept), הצגת יכולות למידה, מקוריות ונזהרנו פעמים רבות שלא לגלוש לפן העסקי של הפרוייקט.