**  
  
  
  
CS 353**

**Database Systems  
Project Proposal  
  
Project Name: Restaurant Express**

**Online Restaurant Order System**

Group 35

Yılmaz Berkay Beken – 21400294

Semih Teker - 21300964

Halil İbrahim Çavdar - 21501613

Bahadır Durmaz - 21502098

1. Introduction

Our assigned CS353 project is to design a data management system for an online food ordering and delivery service. In “Restaurant Express” we provide the user with the ability to search and order for food to be delivered directly to their address. Our system will also provide some innovations such as real time food tracking, easy signup options, automatically determining which district the user resides in, etc...

In this proposal we describe the project in detail through functional, non functional requirements and limitations. We will provide a full ER diagram and dismantle its entity and relationships to explain the attributes and functions one by one.

1. Usage Of The System
   1. Functional Requirements
      1. **Signup**

* Users first have to sign up to use the service.
* Once users sign up their account information will be stored in our database.
  + 1. **Login**
* Users will login to their account by writing their ID and password.
* Users with no recent activity will be logged off automatically and have to re-login.
  + 1. **Log off**
* Users have the option to log off.
* Users will be logged off automatically due to inactivity.
  + 1. **Detect location of user**
* User chooses approximate location for to determine districts of the restaurants.
* User can also enter their location manually or by a map.
  + 1. **List nearby restaurants online**
* Restaurants will be listed to the user according to user’s location.
* Restaurants and their info will stored and accessed from the online database.
  + 1. **Order food**
* Users can order food using our online Restaurant Express service.
* All user orders will be stored and can be accessed by the individual user from our database.
* User will be able to leave notes for the restaurant (such as “no service needed”) while ordering.
  + 1. **Restaurant profile**
* Each restaurant will have a dedicated page, which will be listed, where users can see info such as working hours, menu, rating, most ordered food and drinks…
* Restaurant info will be stored and queried from the online database.
* Each restaurant page will also feature their top selling food and beverages.
* Every ordered food will be counted and stored in the database to form the top products list per restaurant.
* Each restaurant will also feature a ETA (Estimated Time Arrival) for orders derived from the address of the user and restaurants preparation average preparation time.
  + 1. **Rate and review**
* User will be able to rate and review a restaurant’s service for the food they ordered.
  + 1. **Search and filter restaurants**
* User will be able to search for a specific restaurant serving to their district.
* User will be able to filter restaurants for the category of foods they serve. For example if the user wants dessert, mostly bakeries will be listed whereas normal restaurants serving desserts will also be listed but when
  + 1. **Payment**
* User will have multiple payment options such as credit card, cash, online payment for ordering food.
* If paid by Online payment, the info will be stored in the database for later orders.
  + 1. **Joker Sales**
* Restaurants that want to promote can request joker deals.
* Joker deals will be suggested to users that have the tendency to click to joker deals (each joker deal clicks will be stored) like the particular type of food (derived from past orders) the promoting restaurant offers and is inside their district.
  + 1. **Access and reorder from past orders**
* Users will be able to see their past orders and ratings.
* Users can reorder their past order with just one click.
  + 1. **Scheduled ordering**
* User has the option to schedule order to a certain address.
  + 1. **Real-time food tracking**
* User will be able to track the state of the order real-time from when it’s prepared and given to our delivery agent.
* Tracking info and delivery time will be used to asses the average time of preparation and delivery for restaurants as well as the speed of the delivery agents.
  1. Non-Functional Requirements
     1. **User friendliness**
* Our system would be easy to be useful for the users by our filters and user-friendly interface.
  + 1. **Capacity**
* There are various types of the food, restaurants, user and credit card information on our system. Because of that reasons the capacity of the system must be large.
  + 1. **Security**
* Users have to enter their password for logging in to our system to order food. Thus, their credit card info can’t be used by other people.
  1. Limitation
* Users must not be able to cancel their orders after 2 minutes of the order.
* User must not be able to order from restaurants which are out of the distincts from user’s location.
* Users must not be able to access the districts of the restaurants.
* Users must not be able to order before sign in or sign up to the system.
* Users must not be able to order before completing the payment procedure.
* Users must not be able to write comments to restaurants if they had not ordered from this restaurant before.
* Users must not be able to rate the restaurants if they had not ordered from this restaurant before.
* Restaurants must not be able to see the credit card information of the users.

1. Database System

Having a database for our food ordering service is integral since all user info (names, passwords, payment...) and all restaurant attributes (district, location, name, rating, menu…) must be privately stored and be easily fetched and displayed when required.

Our Database design will be fully shown as a ER diagram in the conceptual design part whereas all relations and entities will be dismantled and explained in this part.

* 1. Delivery\_Staff
     1. staff\_id : Unique id of the staff.
     2. source\_address : Address of the restaurant.
     3. destination\_address : Address of the user’s location.
  2. Restaurant
     1. name : Every restaurant has a unique name.
     2. menu : Menu of the restaurant.
     3. address : Address of the restaurant.
     4. {phone\_numbers} : Phone number(s) of the restaurant.
     5. sending\_districts : Districts for servicing to the user which is determined by the restaurants.
     6. working\_hours : Every restaurant has working hours which is for the status for the availability of ordering.
     7. received\_comments : Received comments are stored in database.
     8. history\_of\_orders : Order history will be stored in database.
     9. rating : User’s will rate for the order and restaurant will receive a rating upon that.
  3. Menu
     1. food\_name : Food has a unique name.
     2. food\_type : Food has various types. (Meal, dessert, drink)
     3. price : Prices of the foods will be displayed in
  4. User
     1. ID : User has a unique ID which is given by our system randomly.
     2. name : User has a name for signing up to the system.
     3. password : User has a password for signing in to the system.
     4. email : User has a email address for signing up to the system.
     5. phone\_number : User has telephone number(s) to be able to communicate with restaurant and for signing up for the system.
     6. addresses: User has address(es) for to determine the where the order will arrive by the restaurant.
     7. history\_of\_orders : History of orders will recorded to our system’s database.
  5. user\_address
     1. address\_type : Type of the address(e.g. home, office). Primary key for user\_address is the combination of user ID and address\_type.
     2. Street\_number,street\_name,city, zip-code :Information about the address
  6. Order :
     1. ID : Order ID is a unique number for every order.
     2. date : Date of the order will be recorded to our system.
     3. ordering\_notes : User’s special request(s) will be recorded at ordering notes.
     4. payment\_type : The payment type of the order that user wants.
     5. deliveryTime(): ETA of the order which is calculated by our system by location of the user and restaurant.
  7. District
     1. ID : District has a unique ID.
     2. district\_name : Location name of the user.
     3. area : Boundaries of the district.

1. Conceptual Design

(In the last page of the report.)

1. Website

https://github.com/git-bahadir/Restaurant-Express