Functional requirements

Name:	R. #1. Assign turns at the time of admission
Summary:	Assign turns to users with name and ID data
Input:	String name, int id
Results:	Turn assigned to user
	Incomplete data, turn not assigned

Name:	R. #2. Place a user in the queue
Summary:	Place a user in the queue. There are two queue, the customer and the priority
Input:	
Results:	User located in queue

Name:	R. #3. Search a user in data base
Summary:	Search efficiently a user in data base to get information about user before it reaches the office where it will be attended
Input:	int id
Results:	User was found
	User wasn’t found

Name:	R. #4. Change the amount of the customer’s saving account
Summary:	Change the amount of the customer’s saving account when requesting a withdrawal or consignment
Input:	int id
Results:	Changed the amount of the customer
	User does not have a saving account

Name:	R. #5. Cancellation of a customer’s account
Summary:	Delate client data from database and add it to an exclusive datbase for those who desert
Input:	
Results:	Changed the amount of the customer
	User does not have a saving account

Name:	R. #6. Card payment
Summary:	User can pay amount used by credit card until now. Payment can be by cash or saving account
Input:	int id
Results:	Paid card
	User does not have debts

Name:	R. #7. Undo functionality
Summary:	Undo functionality works to fix mistakes, even after save those
Input:	int id
Results:	Paid card
	User does not have debts

Non-functional requirements

- Implement a ordering method for each of the chosen parameters, with the restriction that only one of them can have averge temporal complexity of n^2
- Users information must be displayed in a spreadsheet-type table
- Must have a graphical interface to display users information and make operations, spreadsheet-type table and status of queues