

2011

We need to design a contact framework which we can use to write contact applications that allow us to manage our contacts. The framework has the following requirements:

- We should be able to add new contact, delete contacts and search contacts.
- For every contact we need to store its name, phone, email, fax number, company name, company phone, street, city and zip code.
- The framework should also support the functionality to show the list of the last 10 executed actions (“add contact” actions and “delete contact” actions), and we should be able to select one of these actions and undo the selected action. Search actions are not shown in this list.

Question 2 [40 points] {50 minutes}

First we have to design a car rental application for one of our customers with the following requirements:

- The application keeps track of which customers (name, address) rents which car (licenceNumber, brand, type, price_per_day) on which days.
- The application keeps track of the list of credit cards that a customer may have
- The application keeps track of payments(amount, date)
- Car rentals can only be paid with credit card.
- Cars are categorized in categories (economy, business(standard, full size, specialty), minivan, suv, etc)

Question 3 [30 points] {30 minutes}

We want to design a points award framework that allows us to write applications that records the number of points in a points award program like frequent flyer miles programs or hotel reward programs. The framework should support the following requirements:

- Members should have a certain status, like bronze, silver and gold, and depending on the status you get a certain number of bonus points
- The possible statuses are application specific
- Every status knows its next status
- Customers can subscribe themselves to notifications on certain events, for example when their account state changes (from silver to gold for example) or when new points are added to their account. Customers can decide themselves how they want to be notified, for example by email or by SMS. The framework should support both notifications by email and notifications by SMS.
- It should be easy to add more notification options, for example by regular mail.
- The history of added points should be available. We should be able to see how many points are added at what date.

2012

Question 1 [40 points] {50 minutes}

Design a general contact framework. This framework should contain the following functionality

- You should be able to **add** and **remove** contacts
- For every contact, the framework should store the first name, last name, email and phone number
- The frameworks can provide a list of the last 20 add or remove contact actions the user has done. We should be able to undo and/or redo any of these.
- Using the framework, it should be very easy to execute different functionality whenever we add or remove a contact. For example, the application that uses the framework might want to send a welcome SMS for every contact we add, and a goodbye SMS to every contact we remove. Or the application writes to a logfile whenever we add or remove a contact.

Now we want to use this contact framework to implement a contact application. This contact application has the following additional requirements:

- The application that uses the framework sends a welcome email to every contact we add, and a goodbye email to every contact we remove.

Question 2 [50 points] {60 minutes}

We need to design a rental framework which can be used to write rental applications such as a DVD rental application or a car rental application. The framework has the following requirements:

- We should be able to add new rentals, delete rentals and search rentals.
- For every rental we need to store the name, phone, email, street, city and zip code of the customer. We also need to keep track of the start date of the rental, the maximum duration of the rental and the end date of the rental.
- The framework should also support the functionality reserve products that you can rent. For every reservation you make, you need to keep track of the date you reserved it.
- The framework should also allow for categorizing the products that you can rent. So you can have for example a category “new releases” for a DVD rental application, and this category contains sub-categories like “drama” or “romance”
- The framework should also support different algorithms to compute the rental fee.
- The framework should support functionality for handling the events when customers don’t return their products on time. Every rental has a start date and a maximum duration. If the maximum duration has passed, the rental application should be able to handle this by sending an email, sending an SMS, writing to a log file, etc. The framework should have the functionality that an event is generated when the

2013-JULY

Question 1 [40 points] {55 minutes}

Suppose you need to make a relatively simple CRUD (Create, Read, Update, Delete) application that manages employees. The application has the following requirements:

- You can add new employees, update existing employees, remove employees, view an employee, search employees and get a list of all employees.
- You should be able to undo/redo the add, update and delete action
- All employees are stored in the database
- You want to log all add, update and delete actions in a logfile

Just when you want to start with the design of this application you get a request from another customer to write a CRUD ProductService application that manages products.

Question 2 [40 points] {45 minutes}

Suppose we need to design a library system framework which allows us to manage Books, borrowings and reservations. Consider the following requirements for this framework:

- The framework should record customer information: name, phone, email, street, city, zip.
- The framework should record book information: title, isbn, author.
- When a customer checks out We should be able to record (checkout-date, return-date, fee) and handle the different books that a customer borrows.
- We should be able to record and handle the reservations of a customer.
- It should be easy to plugin different algorithms to compute the fee for books that are returned to late.
- We can have multiple copies of the same book title. Every copy has a unique scancode.
- The framework should support the ability to create different book categories and subcategories. For example, we can have a category Computer book with subcategory UML books

Question 3 [15 points] {10 minutes}

Consider the following application that uses a dynamic proxy:

```
public interface IVehicle {
    void start();
}

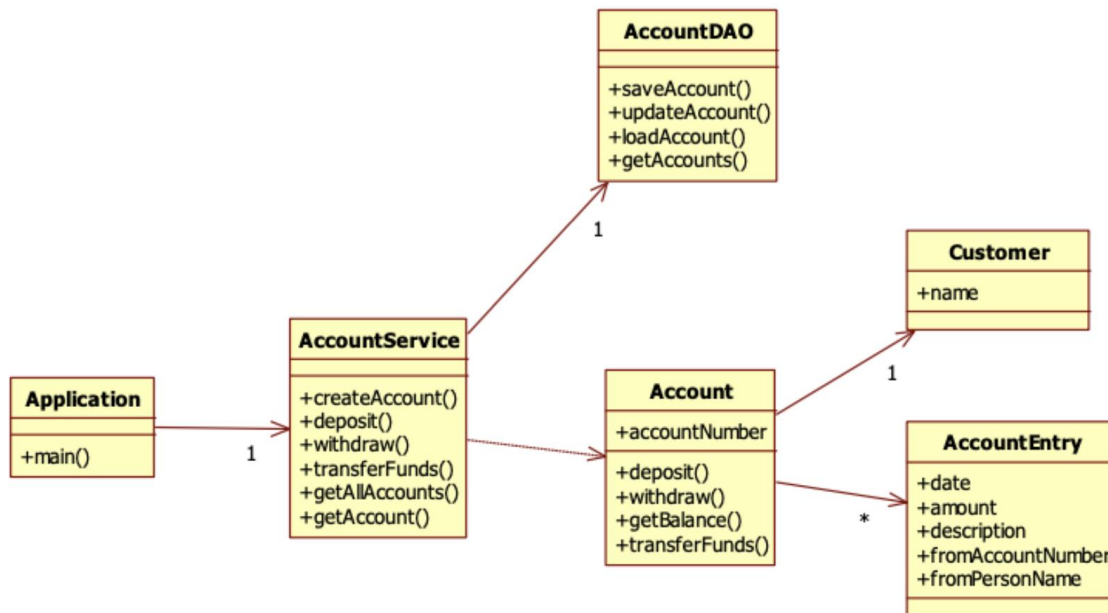
public class Car implements IVehicle {
    private String name ="Herbie";

    public void start() {
        System.out.println("Car " + name + " started");
    }
}
```

2013-MARCH

Question 1 [65 points] {70 minutes}

In a number of design pattern labs in this course, we have applied different patterns to the following given bank application:



Question 2 [15 points] {20 minutes}

Suppose we need to design a drawing framework which allows us to develop drawing applications. Consider the following subset of requirements for this framework:

- The framework should support drawing simple shapes like lines, circles and text.
- We should also be able to move a shape to a different position
- The framework should support grouping and ungrouping of shapes. So we can group a number of shapes together, and then handle this group as one shape. We could for example move this grouped shape to another position on the drawing.

Draw the class diagram that shows how your design works. Do NOT draw the whole class diagram of the drawing framework, but only the class diagram that shows how your design implements the requirements given above. Make sure your class diagram contains all the important information to communicate your design.

Question 3 [15 points] {20 minutes}

Suppose we need to design a game framework which allows us to develop different games. All the games we write contain different levels. The points you receive during the game is a complex formula based on the current level and on the current number of points. The games we develop with this framework can contain any number of levels and different formulas to compute points you receive during the game.

Draw the class diagram that shows how your design works. Do NOT draw the whole class diagram of the game framework, but only the class diagram that shows how your design implements the requirements given above. Make sure your class diagram contains all the important information to communicate your design.

2013-NOV

Question 1 [50 points] {55 minutes}

We have to design a car rental application for one of our customers with the following requirements:

- The application keeps track of which customers (name, address) rents which car (licenceNumber, brand, type, price_per_day) on which days.
- The application keeps track of the list of credit cards that a customer may have
- The application keeps track of payments(amount, date)
- Car rentals can only be paid with credit card.
- Cars are categorized in categories (economy, business(standard, full size, specialty), minivan, suv, etc)
- Customers can become a member of the Rental Rewards Program. Customers receive rental points for every rental. The calculation of how many points a customer receives for a rental depends on the car type and how many days the car is rented.
- Whenever a customer becomes a member of this Renter Rewards Program, the customer gets a Regular Rental Rewards account. When the customer has more than 50.000 rental points, he or she gets an upgrade to a Medium Rental Rewards account. When the customer has more than 150.000 rental points, he or she gets an upgrade to an Ultimate Rental Rewards account.

Question 2 [40 points] {55 minutes}

Design a hotel room reservation system with the following requirements:

- Customers can browse rooms
- Customers can search rooms
- Customers can reserve a room
- Customers can cancel a reservation
- Customers can change a reservation
- Customers can view their reservation
- Rooms are stored in the database
- Reservations are stored in the database
- Whenever a customer reserves a room, the system sends an email to the customer.
- It should be easy to change the technique of sending a message to the customer (SMS, WhatsApp message, etc)
- It should be easy to switch to any type of database, even non relational databases
- Whenever this application sends a message, we want to log that in a logfile
- It should be easy to log to any medium (comma separated file, database, XML file,...)
- We will be using the Spring framework to implement this application
- Every reservation has a unique reservation number
- For every reservation, the system records the start date, end date, total price, number of guest, room, customer first name, last name, email, address info and credit card info.

2014

Question 1 [40 points] {55 minutes}

Suppose you have to design a seat reservation system for a movie theatre with the following requirements:

The seat reservation system covers all cinemas in the country. Cinemas have one or more rooms, and each room contains a set of seats. People can use the system to find particular movie-sessions(shows) by various search criteria like city, time, cinema, movie title, movie category and so on. Movies can be categorized into different categories like for example:

New releases -> Family -> Animation -> ...

Best sellers -> Drama -> ...

Question 2 [25 points] {15 minutes}

Consider the following given application:

```
public class Person {
    private String name;

    public Person(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}

public interface IPersonDAO {
    Person getPerson(String ssn);
}
```

Question 3 [30 points] {40 minutes}

We want to design a points award framework that allows us to write applications that records the number of points in a points award program like frequent flyer miles programs or hotel reward programs. The framework should support the following requirements:

- Members have a certain type of Account, like bronze, silver and gold, and depending on the account type you get a certain number of bonus points
- The possible accounts are application specific
- When your account reaches a certain number of points, you will be upgraded to a higher account type where you receive more points.
- Customers can subscribe themselves to notifications on certain events, for example when their account state changes (from silver to gold for example) or when new points are added to their account. Customers can **decide themselves** how they want to be notified, for example by email or by SMS. The framework should support both notifications by email and notifications by SMS.
- It should be easy to add more notification options, for example by regular mail.
- The history of added points should be available. We should be able to see how many points are added at what date.

2015-OCT

Question 3 [35 points] {45 minutes}

Suppose you need to design a calculator framework that allow us to calculate values like a normal calculator.

We have the following requirements for the **calculator framework**:

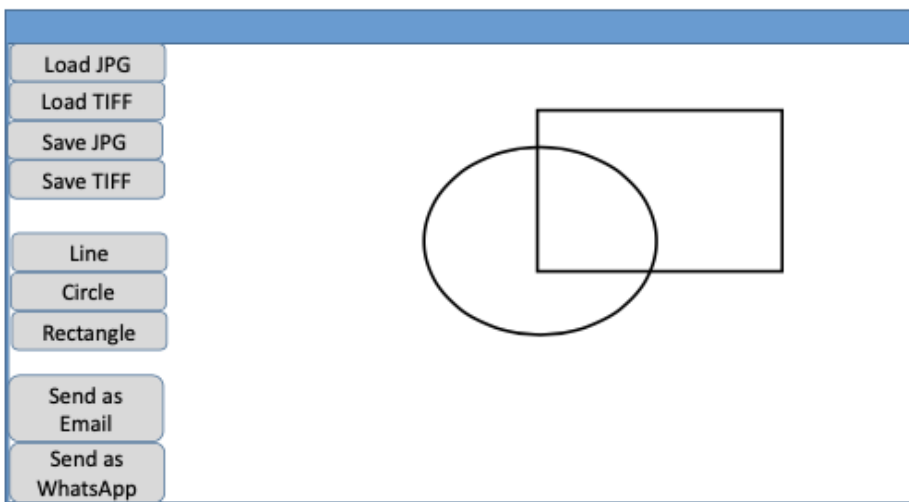
1. The framework should support add and subtract operations
2. The framework supports undo/redo functionality
3. It should be possible to see the list of all calculations you have done on the calculator including the current date, the calculator value (for example 12), the operation (for example "add"), the operation value (for example 5) and the result of the calculation. (in this example 17)
4. The framework will write all calculations that you do in a file on the filesystem
5. It should be easy to add other listener classes that need to do something with the calculator value when it changes.

With this framework we need to design an application that uses this framework. This application has the following requirements:

1. The application also supports multiply and divide operations (next to add and subtract operations)
2. The application will write all calculations that you do in the database.

Question 4 [35 points] {45 minutes}

Suppose you wrote the following paint application:



2015

Question 1 [20 points] {20 minutes}

- a. Explain what Dependency Injection is.
- b. What is the advantage of dependency injection?
- c. Give an example of how we implement dependency injection in the Spring framework. Write code snippets that clearly show how we implement dependency injection in the Spring framework.

Question 2 [20 points] {20 minutes}

- a. Explain what AOP is.
- b. What is/are the advantage(s) of AOP?
- c. Give an example of how we implement AOP in the Spring framework. Write code snippets that clearly show how we implement AOP in the Spring framework.

Question 3 [15 points] {25 minutes}

A customer wants us to design a tool rental application with the following requirements:

1. The application keeps track of which **customers** (name, address, country, phone, email) rent which **tools** (toolnumber, toolname), **price_per_day, price_per_3days, price_per_week, price_per_month,etc**) on which days.
2. Every tool has different pricing. For example a certain drill costs \$10 per day, but if you rent it 3 days, it costs \$20. If you rent it the whole week, it costs \$30. For the whole month it costs \$50. A certain saw might cost \$30 if you rent it 4 days.
3. A customer can rent out **multiple tools** in one **rental**.
4. The application keeps **track of payments**(amount, date)
5. Rentals can be **paid** with credit card, cash, check or with bank wire. If you pay by check, the application should store the name on the check, the check number and the bank name on the check. If you pay with bank wire, the application should store the name of the bank.

Question 4 [20 points] {20 minutes}

Another customer wants us to design a car rental application with the following requirements:

1. The application keeps track of which customers (name, address) rents which car (licenceNumber, brand, type, price_per_day) on which days.
2. The application keeps track of the list of credit cards that a customer may have
3. The application keeps track of payments(amount, date)
4. Car rentals can only be paid with credit card.
5. Cars are categorized in categories (economy, business(standard, full size, specialty), minivan, suv, etc)

We decide to make a generic rental framework.

The framework should support the following requirements:

1. The framework keeps track of which customers (name, address) rents which rental-products on which days.
2. A customer can rent out multiple rental-products in one rental.
3. The application keeps track of payments (amount, date)
4. Rentals can be paid with different payment options
5. Rental-products are categorized in categories and subcategories
6. The framework supports all possible ways to specify the rental price: price per hour, price per half day, price per day, price per week, price per month, price per year, price per 2 weeks, etc.

Question 5 [20 points] {25 minutes}

Suppose we need to design a Game which contains multiple levels. You start in level1, and when you collected enough points, you will move up to level2. For some levels, we use the same algorithm to compute the points you can win, but other levels use another algorithm to compute the points you can win. It should be very easy to add new levels to the game.

- a. Draw a simple UML class diagram of your design for this game.
- b. In a UML sequence diagram show how you advance from one level to another level.

Make sure you add all necessary UML elements to communicate the important parts of your design.

2016

Question 1 [30 points] {40 minutes}

Your company writes software for different **airlines**, and airline A requests you to develop an Frequent Flyer Program application with the following requirements:

There are 2 types of **accounts**, “silver” and “gold”.

Everyone starts with a “silver” account.

When you have more than 10.000 miles or more than 15 flights, you are upgraded to a “gold” account.

Silver accounts receive the same number of miles as the actual miles of their flights.

Gold accounts receive 2 times the number of miles as the actual miles of their flights.

Another airline, airline B, also wants a Frequent Flyer Program application, but it wants more **account types** (bronze, silver, gold, platinum) and it has **different business rules** regarding upgrades and bonus miles.

It is very likely that other airlines also want a Frequent Flyer Program application.

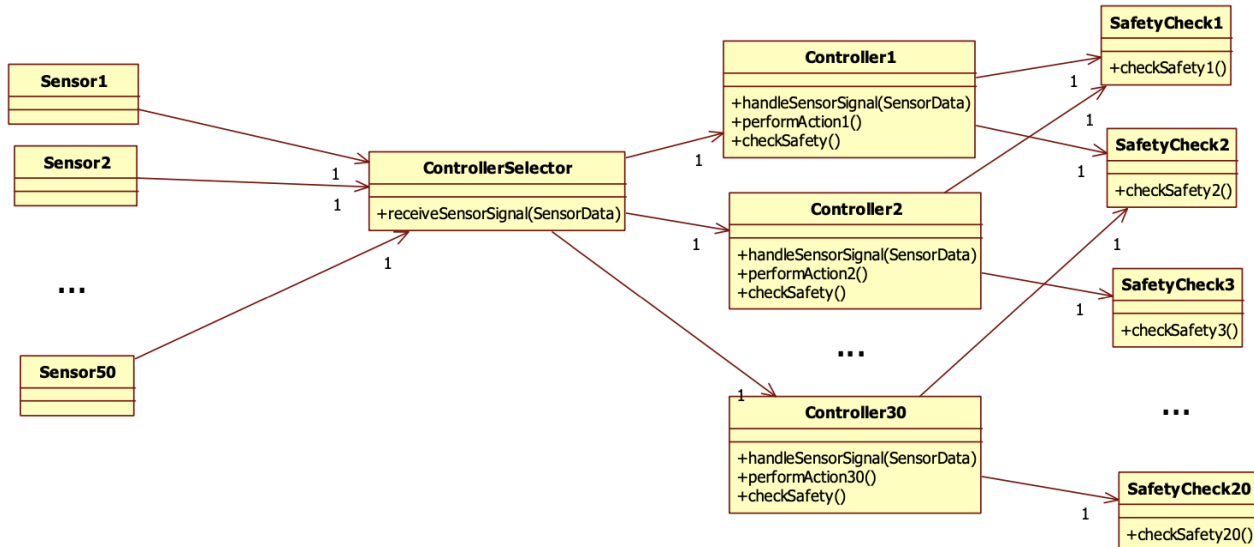
Question 2 [35 points] {40 minutes}

Three years ago we wrote a library application with the following business requirements:

- The application should record **customer information**: name, phone, email, street, city, zip.
- The application should record **book** information: title, isbn, author.
- The application records the **borrowing** of every book. It **records** the checkout-date when the customer takes a book home and it records the return-date and computes the possible fee when the customer returns a book.
- When a customer checks out a book we should be able to **record** (checkout-date, return-date, fee) and handle the different books that a customer borrows.
- We should be able to **record** and handle the **reservations** of a customer.
- We can have multiple copies of the same book title. Every copy has a unique scancode.

Question 3 [30 points] {40 minutes}

Suppose your company needs to design the software for the newest electrical car from Toyota. One of your colleagues came up with the following design:



final

Question 1 [50 points] {60 minutes}

In this DE course we used the message forum functionality of Sakai. In the Sakai forums section you are able to create new forums, delete forums and view forums. For every forum you can create, delete and view topics. For every topic you can create, delete and view threads. For every thread you can create, delete and view messages. You can reply to messages and you can add multiple attachments to messages.

We need to design a general Message-Forum framework. This framework should contain the following functionality

- In the Sakai forums we have a fixed structure where we have multiple forums that contain multiple topics that contain multiple threads that contain multiple messages. The message forum framework should support more flexibility. One application might support only 1 forum with multiple threads, and another application might support multiple forums that contain multiple sub-forums, that contain multiple threads, etc.
- We should be able to **undo and/or redo all** add or delete actions on forums, topics, threads, sub-forums, etc.

2019

Question 1 [20 points] {30 minutes}

Given is the following Package class:

```

public class Package {
    private int length;
    private int width;
    private int height;
    private int weight;

    public Package(int length, int width, int height, int weight) {
        this.length = length;
        this.width = width;
        this.height = height;
        this.weight = weight;
    }

    public int getLength() {
        return length;
    }

    public void setLength(int length) {
        this.length = length;
    }

    public int getWidth() {
        return width;
    }

    public void setWidth(int width) {
        this.width = width;
    }

    public int getHeight() {
        return height;
    }
}

```

Question 2 [35 points] {50 minutes}

Your company writes software for different airlines, and airline A requests you to develop an Frequent Flyer Program application with the following requirements:

There are 2 types of accounts, “silver” and “gold”.

Everyone starts with a “silver” account.

When you have more than 10.000 miles or more than 15 flights, you are upgraded to a “gold” account.

Silver accounts receive the same number of miles as the actual miles of their flights.

Gold accounts receive 2 times the number of miles as the actual miles of their flights.

Another airline, airline B, also wants a Frequent Flyer Program application, but it wants 3 account types: basic, medium and advanced. It has also different business rules regarding upgrades and bonus miles.

It is very likely that other airlines also want a Frequent Flyer Program application.

You decide to write a generic Frequent Flyer Program framework with the following requirements:

Question 3 [20 points] {20 minutes}

In the lab we wrote our own test framework. Suppose all classes we want to test are annotated with **@Service** annotation:

```
public interface Calculator {
    public void reset() ;
    public int add(int newValue);
    public int subtract(int newValue);
}

@Service
public class CalculatorImpl implements Calculator {
    private int calcValue=0;

    public void reset() {
        calcValue=0;
    }

    public int add(int newValue) {
        calcValue=calcValue+newValue;
        return calcValue;
    }

    public int subtract(int newValue) {
        calcValue=calcValue-newValue;
        return calcValue;
    }
}
```


Question 4 [20 points] {20 minutes}

- a. Given is the following code:

```
public class Application implements CommandLineRunner {

    private CustomerService customerService = new
                                   CustomerService();

    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }

    public void run(String... args) throws Exception {
        customerService.addCustomer(new Customer("Frank Brown",
                                                "fbrown@gmail.com"));
    }
}

public class Customer {
    private String name;
    private String emailAddress;

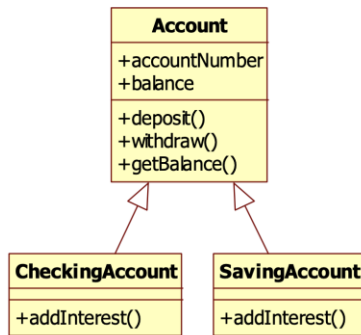
    public Customer(String name, String emailAddress) {
        this.name = name;
        this.emailAddress = emailAddress;
    }
    // getter and setter methods are not shown
}
```

FINAL1

Question 1 [20 points] {20 minutes}

Suppose we need to design a bankaccount framework that support saving accounts and checking accounts. One of the requirements is that we can add interest to every account. The formula to compute the interest is different for saving and checking accounts.

One colleague comes with the following design:



Question 3 [15 points] {20 minutes}

- a) Which pattern that we studied is used by the Spring framework to implement AOP? Only give the name of the pattern.

Proxy

- b) Explain why Spring beans need to implement an interface when we want to make use of the power of dependency injection.

If we want to inject class B instead of class A, then A and B need to be of the same interface type.

- c) Explain why Spring beans need to implement an interface when we want to make use of AOP in Spring.

Spring uses proxy based wiring. If we apply AOP to class A, then the proxy needs to implement the interface of class A.

- d) Give 2 advantages and 2 disadvantages of Spring AOP

Advantages

1. Separation of concern principle
2. Don't repeat yourself principle

Disadvantages:

1. The code is less transparent. It is not clear when some code is executed
2. It is easy to make mistakes
3. Works only for general logic that is the same everywhere

Question 4 [50 points] {70 minutes}

Suppose you have to design an expense tracking framework. The framework allows us to keep track of all our financial transactions, both income and expenses. The expense tracking framework has the following requirements:

- We should be able to add new financial transactions
- We should be able to remove existing financial transactions
- We should be able to update existing financial transactions
- For every transaction, we want to record its name, data, amount, category name and transaction type (income or expense)
- Transactions can be placed in categories, for example groceries, fuel, insurance, car, medicine, etc.
- We should be able to add a new category
- We should be able to remove an existing category
- We should be able to update an existing category
- For every transaction, we want to record its name and a short description
- All categories and transactions should be stored in a database
- We have a GUI page that allows us to add, remove or update categories
- We have a GUI page that allows us to add, remove or update financial transactions