Financial Control Specs
Bernardo Bicalho

---

1 Introduction

Poor financial management can complicate someone's life in many ways. For example, they may not be able to afford their dream house or car, because they spend all their money on weekends or are simply clueless about how much they are spending. However, that's the least of it, there are major issues such as getting into debt with the bank. In such cases, the person may end up with a bad credit record or even never recover from the financial blow due to compound interest.

Aiming to help users bring their financial lives under control, the Financial Control System assists in mapping all their expenses, income sources and investments, through an intuitive, easy-to-navigate interface with customizable charts and colors that help users quickly identify the nature of each of information.

The objective of this project is to present a software that includes a database, back-end and graphical user interface (GUI) for navigation. The integration between these three components is responsible for recording (and compressing), editing, deleting and managing user transactions related to expenses, income sources and investments.

This document presents the mini-world description, requirements analysis, and all the diagrams and necessary scripts to create the proposed database.

2 Mini-world description

Below are the specifications for the FCS (v1.0) mini-world. The FCS (Fincancial Control System) manages the entire transaction flow of a user.

Each user is identified by a code and also includes the name, username, email and password.

Each expense is identified by a code and also includes the type, value, payment method and date.

Each income is identified by a code and also includes the type, source, value, institution and date.

Each investment is identified by a code and also includes the type, value, institution and date.

Each user can record multiple expenses and an expense can be recorded by only one user.

Each user can receive multiple incomes and an income can be received by only one user.

Each user can make multiple investments and an investment can me made by only one user.

3 Requirements analysis

3.1 Functional requirements

FR01: The system must allow the user to create an account
FR02: The system must allow the user to record an expense
FR03: The system must filtering expenses by type
FR04: The system must filtering expenses by payment method
FR05: The system must filtering expenses by date

FR06: The system must allow the user to record an income
FR07: The system must filtering incomes by type
FR08: The system must filtering incomes by instituion
FR09: The system must filtering incomes by date
FR10: The system must allow the user to record an investment
FR11: The system must filtering investments by type
FR12: The system must filtering investments by date
FR13: The system must filtering investments by instituion
FR14: The system must calculate the opening balance
FR15: The system must calculate the end-of-day balance
FR16: The system must sum the total spent in a day, month, year
FR17: The system must calculate the total income for a month
FR18: The system must calculate the monthly profit
FR19: The system must calculate the total invested in a month

3.2 Non functional requirements

NFR01: (Styling) The system shall allow the user to customize colors associated with each transaction type and income source, in order to visually differentiate them throughout the interface

NFR02: (Feedback) The system shall dynamically adjust the background color of financial summary values (expenses, total income, total invested, and total profit) based on thresholds defined by the user, to visually indicate the financial condition

NFR03: (Security) The system shall store user passwords using a secure hash function

NFR04: (Compression) The system shall compress all stored data using the Huffman algorithm

4 Conceptual design

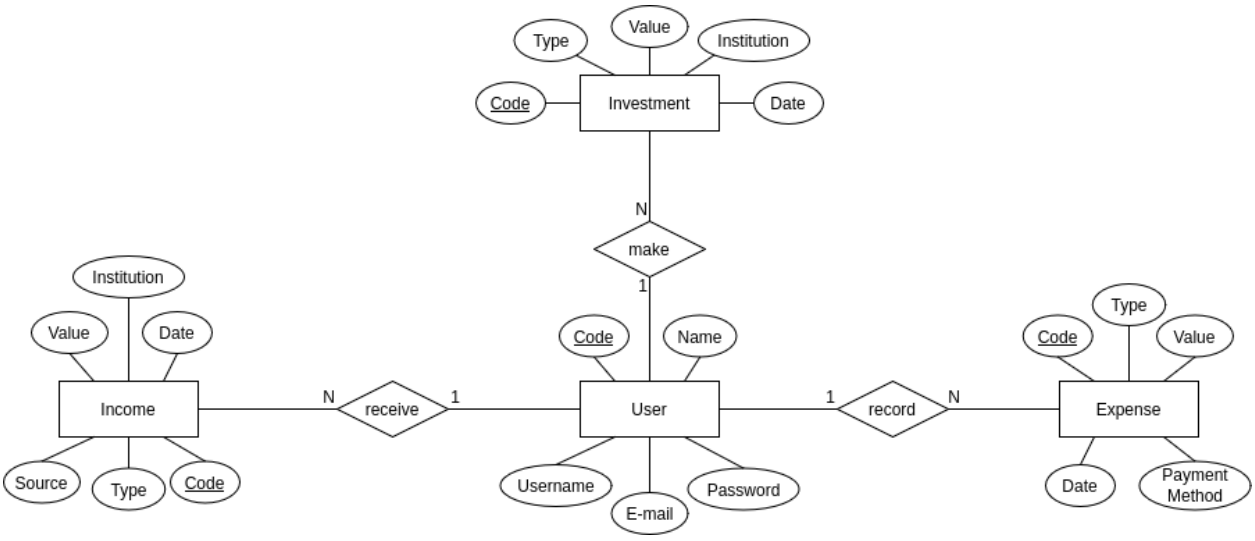

Figura 1: DER: FCS

5 Logical design

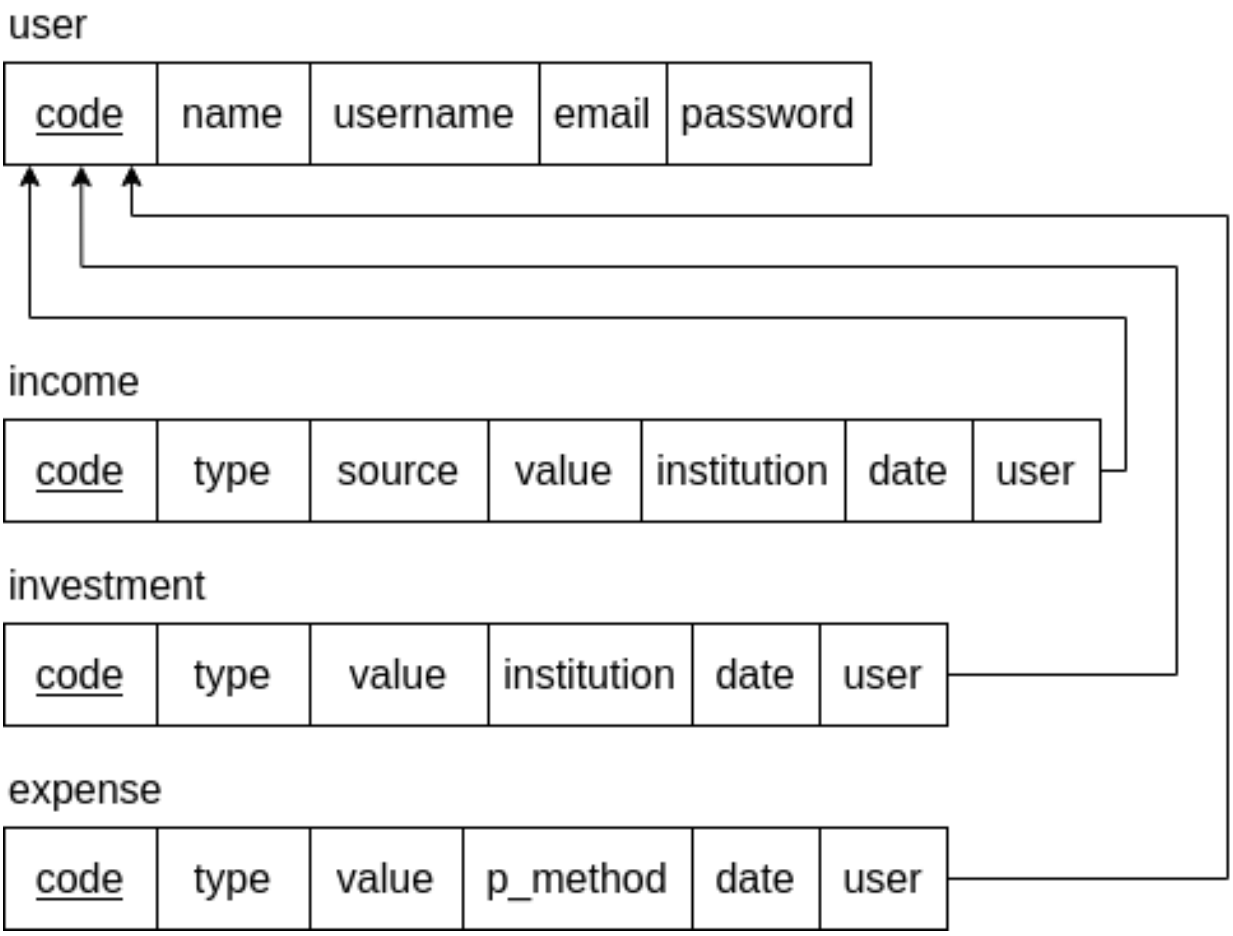

Figura 2: DSC: FCS

6 Physical design

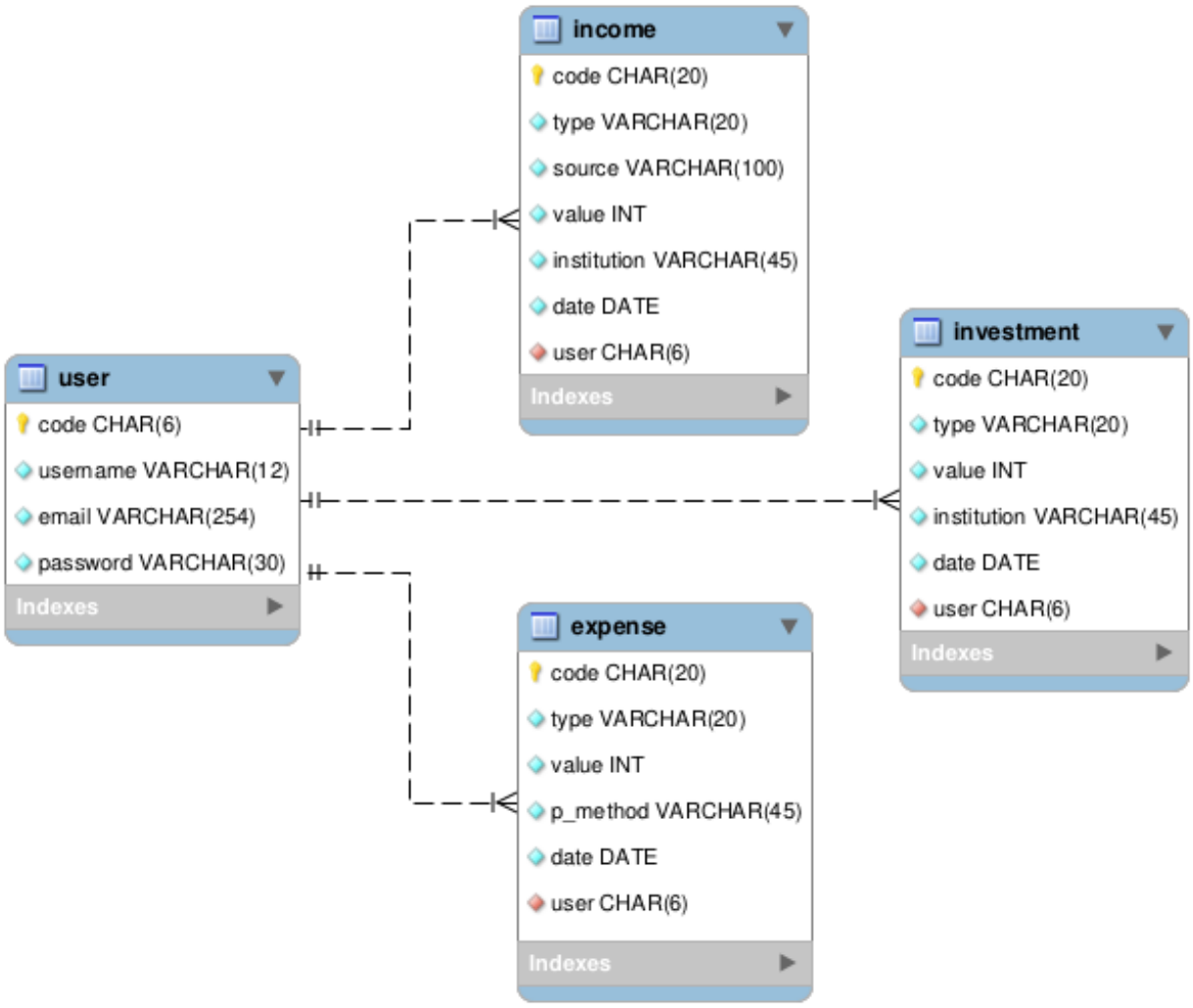

Figura 3: DI: FCS

7 Database setup SQL query

```
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,
    STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,
    ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- -----------------------------------------------------
-- Schema fcs
-- -----------------------------------------------------

-- -----------------------------------------------------
-- Schema fcs
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `fcs` ;
USE `fcs` ;

-- -----------------------------------------------------
-- Table `fcs`.`user`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `fcs`.`user` ;

CREATE TABLE IF NOT EXISTS `fcs`.`user` (
  `code` CHAR(6) NOT NULL,
  `username` VARCHAR(12) NOT NULL,
  `email` VARCHAR(254) NOT NULL,
  `password` VARCHAR(30) NOT NULL,
  PRIMARY KEY (`code`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `fcs`.`income`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `fcs`.`income` ;

CREATE TABLE IF NOT EXISTS `fcs`.`income` (
  `code` CHAR(20) NOT NULL,
  `type` VARCHAR(20) NOT NULL,
  `source` VARCHAR(100) NOT NULL,
  `value` INT NOT NULL,
  `institution` VARCHAR(45) NOT NULL,
  `date` DATE NOT NULL,
  `user` CHAR(6) NOT NULL,
  PRIMARY KEY (`code`),
  CONSTRAINT `fk_user_income`
    FOREIGN KEY (`user`)
    REFERENCES `fcs`.`user` (`code`)
```

```
48      ON DELETE CASCADE
49      ON UPDATE RESTRICT)
50  ENGINE = InnoDB;
51
52  CREATE INDEX `fk_user_income_idx` ON `fcs`.`income` (`user` ASC) VISIBLE;
53
54
55  -- -----------------------------------------------------
56  -- Table `fcs`.`investment`
57  -- -----------------------------------------------------
58  DROP TABLE IF EXISTS `fcs`.`investment` ;
59
60  CREATE TABLE IF NOT EXISTS `fcs`.`investment` (
61    `code` CHAR(20) NOT NULL,
62    `type` VARCHAR(20) NOT NULL,
63    `value` INT NOT NULL,
64    `institution` VARCHAR(45) NOT NULL,
65    `date` DATE NOT NULL,
66    `user` CHAR(6) NOT NULL,
67    PRIMARY KEY (`code`),
68    CONSTRAINT `fk_user_investment`
69      FOREIGN KEY (`user`)
70      REFERENCES `fcs`.`user` (`code`)
71      ON DELETE CASCADE
72      ON UPDATE RESTRICT)
73  ENGINE = InnoDB;
74
75  CREATE INDEX `fk_user_investment_idx` ON `fcs`.`investment` (`user` ASC)
        VISIBLE;
76
77
78  -- -----------------------------------------------------
79  -- Table `fcs`.`expense`
80  -- -----------------------------------------------------
81  DROP TABLE IF EXISTS `fcs`.`expense` ;
82
83  CREATE TABLE IF NOT EXISTS `fcs`.`expense` (
84    `code` CHAR(20) NOT NULL,
85    `type` VARCHAR(20) NOT NULL,
86    `value` INT NOT NULL,
87    `p_method` VARCHAR(45) NOT NULL,
88    `date` DATE NOT NULL,
89    `user` CHAR(6) NOT NULL,
90    PRIMARY KEY (`code`),
91    CONSTRAINT `fk_user_expense`
92      FOREIGN KEY (`user`)
93      REFERENCES `fcs`.`user` (`code`)
94      ON DELETE CASCADE
95      ON UPDATE RESTRICT)
96  ENGINE = InnoDB;
97
```

```
98 CREATE INDEX `fk_user_expense_idx` ON `fcs`.`expense` (`user` ASC) VISIBLE
       ;
99
100
101 SET SQL_MODE=@OLD_SQL_MODE;
102 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
103 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```