A cartoon astronaut with headphones

Description automatically generated



Moroccan School of engineering sciences

**END-OF-YEAR INTERNSHIP REPORT**

# Sector: Computer Engineering and networks

Title:

Comprehensive Financial Management System: “BalanceLab”

By

**Yasser Choubi**

Realized in

A cartoon astronaut with headphones

Description automatically generated

Supervisors:

**Mr. Raissi Driss :** TalkLab Supervisor

**Mr. Ilyas Bentachfine :** Emsi Supervisor

## Acknowledgments

I am indebted to my supervisors Mr. Driss Raissi and Mr. Bentachfine Ilyas for their constant guidance, the support they provided me with during my 2 months of training period was immense something which no amount of words could describe.

During this short time but indispensable one for me, I had the chance to work with them so that my new memories and experiences can be useful in other places from now on. I appreciate the time and attention they have given me. They have motivated me to continue in this field, and I am grateful for the respect they afford me.

Thank you again for all of your help over the last couple of months. Your dedication to my growthand development is deeply appreciated, and I am grateful to have had the chance to work with such knowledgeable and supportive individuals as both of you.

## Abstract

Before the implementation of BalanceLab Financial Management System, the accountants relied solely on paper-based workflows to record the income and the expenses of the company. This manual activity was laborious, and there was a high level of error, miscalculations, and ineffective waste management. Since physical parchments often went missing or got poorly filled, tracking the finances proved tedious and created unnecessary delays.

Now, with BalanceLab coming into use, accountants are able to record expenses and income of the company electronically, and as such, the whole process is made cleaner, quicker and easier to control. The platform serves as the single source of up-to-date operational information without any dependencies of data flows internal or external. This has drastically reduced errors and increased the degree of efficiency in conducting financial activities within and across units.

## Contents

[Acknowledgments 2](#_bookmark0)

[Abstract 3](#_bookmark1)

[Contents 4](#_bookmark2)

[Acronyms 7](#_bookmark3)

[List of figures 8](#_bookmark4)

[General Introduction 9](#_bookmark5)

[Problematic 9](#_bookmark6)

[Proposed solution 9](#_bookmark7)

[Conclusion 10](#_bookmark8)

1. [Chapter 1: Presentation of Host organization 11](#_bookmark7)
   * 1. [Introduction 12](#_bookmark9)
     2. [History[6] 12](#_bookmark10)
2. [Chapter 2: Definition of Requirements 13](#_bookmark7)
   1. [State of the art 14](#_bookmark20)
      1. [Introduction 14](#_bookmark21)
      2. [Conclusion 14](#_bookmark22)
3. [Design and Modeling 15](#_bookmark23)
   * 1. [Introduction 15](#_bookmark24)
     2. [Class Diagram 15](#_bookmark24)
     3. [Use case Diagram 17](#_bookmark24)
     4. [Sequence Diagram “Sign in” 18](#_bookmark24)
     5. [Sequence Diagram “Add Expense” 19](#_bookmark24)
     6. [Sequence Diagram “Update Employe” 20](#_bookmark24)
     7. [Activity Diagram “Register User” 21](#_bookmark24)
     8. [Conclusion 22](#_bookmark24)

[Chapter 3: Used Technologies 23](#_bookmark7)

* 1. [Introduction 24](#_bookmark27)
  2. [Front End languages 24](#_bookmark28)
     1. [Angular[1] 24](#_bookmark29)
     2. [CSS (Cascading Style Sheets)2] 24](#_bookmark30)
     3. [Ng-Zorro[3] 24](#_bookmark31)
  3. [Back-End languages 25](#_bookmark32)
     1. [Java[4] 25](#_bookmark33)
     2. [Spring Boot[5] 25](#_bookmark34)
  4. [Conclusion 25](#_bookmark37)
     1. [Chapter 4: Requirements 26](#_bookmark7)
  5. [Introduction 27](#_bookmark38)
  6. [Functional Requirements 27](#_bookmark39)
  7. [Non-Functional Requirements 28](#_bookmark40)
  8. [Conclusion 28](#_bookmark41)

1. [Chapter 5: Graphical Interpretation 29](#_bookmark7)
   1. [Introduction 30](#_bookmark42)
   2. [Website abilities 30](#_bookmark43)
      1. [Registration Page 30](#_bookmark45)
      2. [Home Pages 32](#_bookmark46)
      3. [Live Charts 33](#_bookmark46)
      4. Incomes Page 34
      5. Expenses Page 35
      6. Create Employee 36
      7. Employee Profile 38
2. [Chapter6: Conclusion and perspective 39](#_bookmark7)

[GENERAL CONCLUSION 40](#_bookmark58)

* 1. [Introduction 41](#_bookmark59)
  2. [Perfection Constraints 41](#_bookmark60)

#### [Cost classification and prioritization 41](#_bookmark61)

* + 1. [Real-time financial insights and reporting 41](#_bookmark62)
    2. [Employee Cost selection 41](#_bookmark63)
    3. [User Access and data security 41](#_bookmark64)

[Resources 42](#_bookmark66)

## Acronyms

EMSI : École Marocaine des Sciences de l'Ingénieur.

## List of figures

Figure 1: Class Diagram 15

Figure 2: Use case Diagram 17

Figure 3: Sequence Diagram “Sign in” 18

Figure 4: Sequence Diagram “Add Expense” 19

Figure 5: Sequence Diagram “Update Employe” 20

Figure 6: Activity Diagram “Register User” 21

Figure 7:Security Detail 30

Figure 8: Register User 31

Figure 9: Login Page 32

Figure 10: Home Page 32

Figure 11: Income and Expense charts 33

Figure 12: Incomes Page 34

Figure 13: Expenses Page 35

Figure 14: Update Expense 36

Figure 15: Create Employee 36

Figure 16: Schedule Salaries 37

Figure 17: List of employees 37

Figure 18: Employee Profile 38

## General Introduction

This internship has been a crucial part of my journey, as it marks my second practical experience in the field of web application development. Throughout this period, I have had the opportunity to work on the creation of a web application that holds immense potential in the digital realm.

Building a web application requires a keen understanding of various technologies, frameworks, and design principles. It involves ensuring a seamless user experience, efficient data handling, and robust security measures. From the frontend user interface to the backend functionalities, every aspect must align harmoniously to deliver a cohesive and impactful user experience.

In this report, I will delve into the development process of my web application, exploring the architecture, functionalities, and features that make it unique. Through this internship, I have been exposed to the intricacies of web application development, honing my skills and expanding my knowledge base.

## Problematic

* How to address the inefficiencies caused by accountants writing everything down on paper, causing delays and mistakes that make it hard to see a real-time picture of the company?
* How to transition from the paper-based system towards BalanceLab so that this process goes strictly according to plan and does not adversely affect the financial operation of the company?

## Proposed solution

* Detailed Transition Plan: Create a reliable transition strategy with the help of onboarding sessions, training workshops and practical user manuals that project accountants can use while acclimatizing themselves to BalanceLab. This will help them make a more seamless transition from the paper-based exercise, and minimize resistance or inconvenience in running daily operations.
* User Experience: A friendly user interface that reflects normal accounting tasks such as entering expenses and incomes (e.g., the Design must take its User to all possible entry screens intuitively). It is possible to add Still user-friendly tools that make an entry of data easier and creates faster adoption.
* Employee Management Integration: Implement the possibility to add employees in the system that will allow accountants to filter expenses by employers. With this feature, it will be easier to sort and monitor the financial information input, especially when dealing with different departments/ companies, thus improving the system’s efficiency and result.
* Mobile Compatibility: Ensure the BalanceLab application is compatible with mobile devices, allowing accountants to create new data on-the-go using smartphones or tablets. This flexibility will enhance convenience and efficiency in data creation.

## Conclusion

Working on the creation of the BalanceLab Financial Management System I got rich practiced experience and extended the knowledge about FM application creation. This internship has enabled me not only to develop technical skills but also to remind me of my passion of being able to develop solutions that make work easier.

Further, as this report progresses, I will describe the process of creating BalanceLab and explain how it can revolutionize the way businesses approach their balance and explain how it can revolutionize the approach to the balance sheet and demonstrate the benefits of efficiency, increase in accuracy and productivity, and better reporting for Companies. This has been one of the most productive lessons in my career development as a possible in the prospects of this project to transform the management of finances in businesses.

# Chapter 1: Presentation of Host Organization

### Introduction

This chapter is a brief presentation of the Host Organization where I did my internship, as well as the different missions defined by TalkLab.

### History[[6]](#_bookmark67)

TalkLab, originally founded in 2016 under the name "Smart Call Distribution" (SCD), is a customer relationship management and outsourcing company based in Casablanca, Morocco. Over the years, the company rebranded to TalkLab and expanded its services to become a major player in the call center industry. They specialize in inbound and outbound call campaigns, telemarketing, and multichannel customer interaction strategies, supporting various sectors such as e-commerce, telecommunications, and insurance.

TalkLab emphasizes an integrated and multichannel approach to customer engagement, leveraging technologies like geomarketing, internet moderation, and data-driven insights to enhance client relationships. Their growth has been driven by a focus on service quality, efficient call management, and a commitment to fostering strong relationships with clients​.

# Chapter 2: Definition of Requirements

### State of the art

#### Introduction

According to the present practice in the fast-growing area of financial management, innovation has provided a variety of solutions, which are aimed at fulfilling similar functions. BalanceLab which is a system designed for demand management, company expenses, and incomes found itself in a rather saturated market since there are numerous financial management tools available each providing an individual perspective on addressing the concerns.

There are several well-known solutions available in the market, which include QuickBooks, Xero, and FreshBooks, that is designed for specific financiers. QuickBooks also regarded as the powerful accounting software that offers the prolific tools for payroll, invoicing and reporting. Xero, for instance, is easy to use, and fully cloud-based – appeals to the owners of small businesses. FreshBooks aims its solutions mainly to independent contractors and service-oriented companies with a simple to use tool for issuing invoices and tracking the expenses.

BalanceLab sets itself apart from the pack in that the application is more oriented toward the easier entry of company expenses and incomes especially if the organization is coming from a predominantly paper-based environment. They are automation reporting services, ability to enter data in a centralized, and cost tracking according to specific employers needs in an organization. While there are already many existing tools brought forward, BalanceLab is a more flexible solution for the companies that aim to enhance the accuracy and effectiveness of the financial data tracking.

#### Conclusion

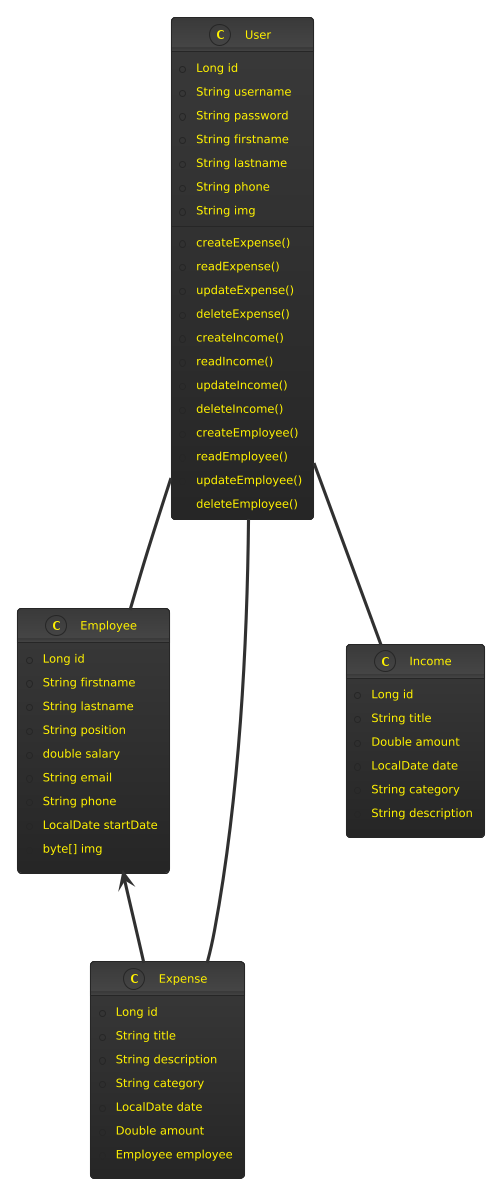
In conclusion, it is evident that there are already similar apps to the one being developed. The digital landscape is constantly evolving, and with each passing day, numerous developers and entrepreneurs are creating applications that cater to similar functionalities and target audiences. The presence of these similar apps should not be seen as a deterrent but rather an opportunity for improvement and differentiation.

### Design and Modeling:

#### Introduction

In the design and modeling phase of my project, we delve into the critical process of translating our conceptual ideas and requirements into tangible representations. This phase plays a pivotal role in building a solid foundation for our application’s architecture and functionality. By employing various modeling techniques and tools, we gain insights into the system’s structure, behavior and interactions allowing us to make informed decisions before the actual implementation.

In this section, we’re going to present a series of diagrams that visualize different aspects of our application‘s design. these diagrams enable us to understand the relationships Between entities, the flow of data and processes. The primary diagrams we included are:



*Figure 1: Class Diagram*

The class diagram of BalanceLab system includes several core entities which are responsible for organizational financial information operations, its personnel and users. The following core entities have been defined for the system: Employee, Expense, Income and User to perform the following roles in the system.

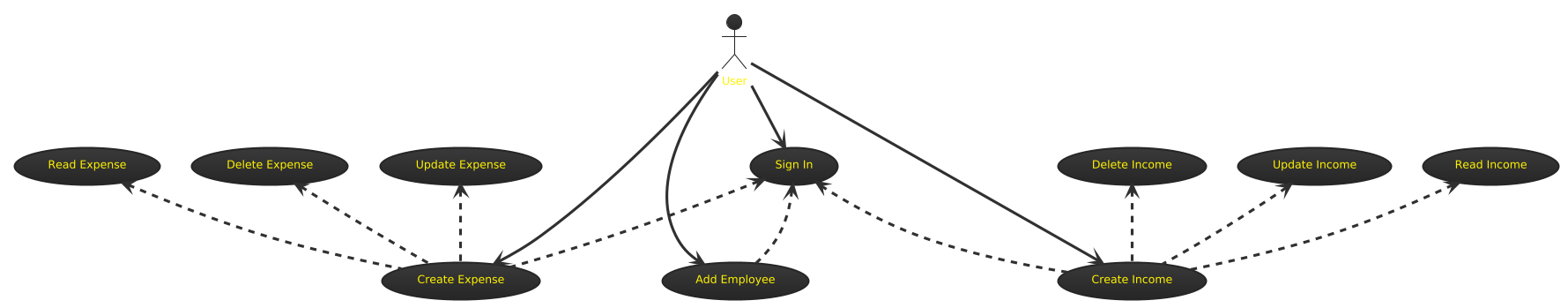
- \*\*Employee\*\*: This entity gives information about the company’s employees such as name, the position he or she occupies, salary, contact details, and date he or she began working for the company. Each employee may have several linked costs, which show expenses of an employee for the enterprise. Heading to the next table is a one-to-many relationship between the `Employee` and `Expense` table as an employee can have one or many expenses assigned to him.

- \*\*Expense\*\*: The `Expense’ entity is used to capture many expenses in the company and some of the attributes they include the title, description, amount, category and the date of the expense. The relationship between each of the expenses and the employee is defined as belonging to same table \*\*many to one\*\* while, Partnership employee Expense has many Employees, but each employee has only one Partnership employee Expense.

- \*\*Income\*\*: This entity is responsible for the processing of revenue or income information of the company. Some of the formats it has include title, amount, date and description of the income incident. Unlike `Expense` which is related to employees, the `Income` entity is an independent entity that is not directly associated with employees.

- \*\*User\*\*: The `User` entity gives an account of the people who will be using the system; these are usually the administrators or the finance personnel. The CRUD operation can work with both expenses and incomes and it is also possible to manage the employees. Every user has his or her own account; each of which contains alphanumeric login details, password, and personal contact data.

The diagram ensures a well-organized structure where the diagram ensures a well-organized, it supports a visually pleasing as well as an efficient “System of Financial Management” to monitor and track the expenses incurred on the employees as well as the total income generating capacity of the company. The mapping of described relationships provide transparency of entities and interactions which make them quite efficient in tracking and managing financial data.

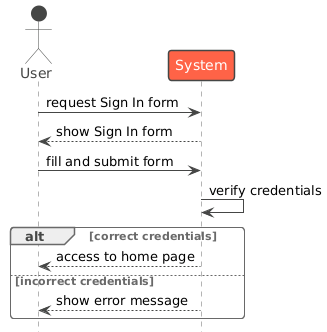


*Figure 2: Use Case Diagram*

In the use case diagram, we get to see the interactions that can occur together with the most basic functionalities accessible to the users. Using the sign-in option, any user can perform a set of the basic financial operations connected with creation, reading, updating, and deleting of both expenses and incomes. Also, there are some other features like the ability to include new employees into system which gives user certain amount of control in the process of managing employees.

The diagram illustrates how these current features are expanded on by the operations of record read, record update, and record delete in order to depict how a user can perform multiple phases of managing financial information. The addition of an expense or income, addition of an employee and other actions all mean that the user has to first login in, thus proving that all the operations are well protected and linked to a given user session ID.

This use case diagram helps to make a clear distinction of the user roles and the actions that are possible with the BalanceLab system, depicting the management of expenses, incomes and employees. It also focuses on the access control since sign-in is mandatory before executing any of those operations.



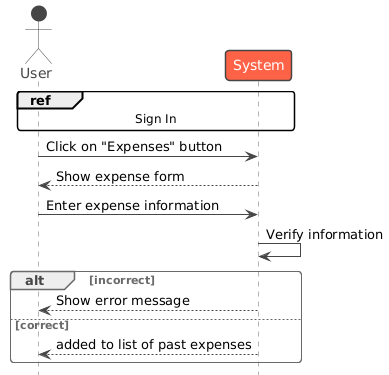
*Figure 3: Sequence diagram “sign in”*

The process begins with a accountant who intends to sign in to the system. First, he must have registered account by the admin, providing their essential details such as username, full name ,phone number and creating a password. Once registered, when the accountant wishes to access the system, they navigate to the sign-in page.

On the sign-in page, the accountant encounters an option to request a sign-in form, presented as a button. This request triggers the system to generate a unique sign-in form tailored specifically for that particular request. The generated form includes fields relevant to the sign- in process, a username and a password.

The accountant proceeds to fill out this generated sign-in form, entering the requested information accurately. Upon submission, the system swings into action, initiating a verification process. This verification process involves cross-referencing the information provided by the accountant with the records in the system's database. Specifically, the system checks if the provided username and password match those on record for the registered accountant.

If the verification process concludes successfully, meaning the provided information is correct and matches what is on record, the system grants access to the accountant, and they are redirected to the home page. However, if any discrepancies or errors are detected during the verification process, the system promptly displays an error message, indicating that their sign-in attempt has failed.



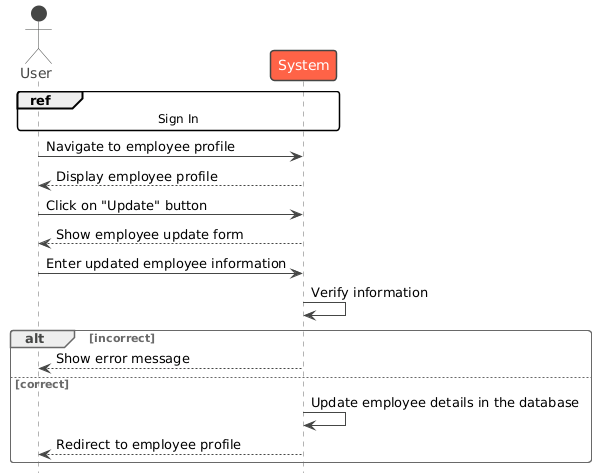
*Figure 4: Sequence diagram “Add Expense”*

Before accruing an expense in the BalanceLab system, the user must log in by providing his or her correct username and password. To enter a new expense, once he or she has authenticated, the user proceeds to the ‘Create Expense’ tab normally accessed through the ‘Expenses’ button which is normally found at the sidebar of the web application.

On clicking the button, the system provides the user with a form where the user has to enter details regarding the expense like the title of the expense, the description of the expense, the category of the expense, date of the expense and the amount of the expense. Some of the fields that the user has to fill with appropriate data include fields that contain information for the system to process the paid expenses.

The user then enters the expense details into the web form and submits it and the system verifies it. The system performs validity check on the given information to confirm if all information that has been requested is complete and accurate. For any of the mandatory fields that are not filled or when there is invalid input, a message appears to the effect that the expense cannot be processed until the problems with the identified fields are sorted.

Yet if all the blasted data entered is valid and comprehensive, then the system goes ahead and insert the expense to the database. After the process is successfully done, we can find the expense in the list of past expenses. This leaves a clear path where users can monitor their companies’ transactions in a very clear format.



*Figure 5: Sequence diagram “Update Employee”*

In order to amend an employee’s details in the BalanceLab system, the user has to log in using proper login details. When the user is authenticated, they will be able to go to the employment section by choosing the specific employee of interest either by list or search options. The system shows the current details of the employee; this way the user can see what has been entered before making modifications.

Then the user can click on Update button which will take them to the update page with all the information of the selected employee filled in the form. This form consists of fields where changes of the employee details, the name, position, salary, contact details, etc. can be made. After this the user just completes the required fields and submits the form.

After submission the system checks all the modified information provided by the user to ensure they are correct and no information is missing. If any fields are left blank or if data entered is incorrect, the system generates an error message to the effect that the information has been entered wrongly.

If all the information is correct the update is made and the new details of the employee is stored in the database. Finally after a successful update, the user is led back to the employee’s profile page so that he or she can see the updated data. This flow as much as enables the upkeep of accurate records of the employees within the system.

#### 

*Figure 6: Activity diagram “Register User”*

The following activity diagram shows how a new user is to be registered in the BalanceLab system securely. First, one is supposed to open the registration webpage and in the process, the system presents a code that is required to be entered by the user. This added form of security makes it impossible for any unauthorized person to fill the registration form.

After the participant submits an access code, the system checks whether the code is correct or not. If the code is invalid, the user receives the corresponding message and get redirected to login page. If the code is valid, then the user is allowed to the registration form whereby they fill some of the crucial information they include username, password, first name, last name, phone number, and profile picture.

Upon the completion of the form, the system does a scan scrutinizing whether the information that the user has keyed-in is accurate or missing. It is impossible to write any minute detail missing or write the wrong version which is corrected by the system. If the details provided are approved by the verification process, the system facilitates the creation of the new user account and the user is taken to the login page which concludes the registration process.

From this diagram, there is a two-stage security procedure, which starts with validation of the access code and then followed by the verification of the user details before valid information can be entered and result in creation of a new account in the system.

#### II.2.3. Conclusion

In conclusion, the design and modeling phase of the project proved to be a crucial step in converting abstract concepts and requirements into concrete representations. Through the utilization of diverse modeling techniques and tools, valuable insights into the system’s structure, behavior and interactions were gained. These insights facilitated informed decision making before the actual implementation, ensuring a solid foundation for the application’s architecture and functionality.

# Chapter 3: Used Technologies

### Introduction

While working on our project, we used effective usage of today’s technologies,this creating functional and interactive web application. For the front-end, we employed “Angular” with “ng-zorro” and “CSS” in order to develop an efficient and overall friendly UI. These technologies enable the proper structuring of content, reacting to the layouts, and giving an intuitive feel to the users with great Angular-based components offered by ng-zorro.

For the back-end we used the “Spring Boot” a popular Java framework designed for the rapid development of the enterprise-grade applications “as the primary foundation for our business logic and for the optimized data management. As Spring Boot initiated the process of application designing and building it provided ideal features such as dependency injection, database integration options and modish architecture to build a prototype.

We were able to build a complete application that achieved the objectives of the project and would offer the end users a good experience while leveraging the front-end flexibility of Angular, ng-zorro, and CSS and the robust back-end power of Spring Boot.

### Front End languages:

#### Angular[[1]](#_bookmark68)

Angular is a platform and framework for building single-page client applications using HTML and TypeScript. Developed and maintained by Google, Angular provides a comprehensive solution for developing robust web applications. It features a component-based architecture, which enhances modularity and reusability of code. Angular includes powerful tools and libraries for routing, form management, HTTP client, and more, enabling developers to build dynamic and responsive web applications efficiently. The framework also supports two-way data binding, dependency injection, and an extensive ecosystem of extensions and libraries, making it a popular choice for large-scale enterprise applications.

#### CSS (Cascading Style Sheets[)[2]](#_bookmark69)

CSS is a stylesheet language employed in conjunction with HTML to control the presentation and layout of web pages. It allows web developers to define styles, such as colors, fonts, margins, padding, and positioning, for HTML elements. By separating content (HTML) from presentation (CSS), developers can ensure consistent styling across multiple web pages and achieve a visually appealing and responsive design for various devices.

* + 1. **Ng-zorro**[**[3]**](#_bookmark70)

Bootstrap is a popular open-source front-end framework developed by Twitter. It is built with HTML, CSS, and JavaScript and provides pre-designed templates, components, and styles to expedite web development. By using Bootstrap, developers can create responsive, mobile-first websites and web applications efficiently, as it takes care of the cross-browser compatibility and consistent UI elements.

### Back-End languages

**3.1 Java**[**[4]**](#_bookmark71)

Python is a high-level, versatile, and interpreted programming language known for its simplicity and readability. Guido van Rossum developed it in the late 1980s. Python supports multiple programming paradigms, including object-oriented, functional, and imperative styles. It is widely used for web development, scientific computing, data analysis, artificial intelligence, automation, and more. Python's extensive standard library and large community of developers contribute to its popularity and ease of use.

**3.2Spring Boot**[**[5]**](#_bookmark72)

Spring Boot is a powerful framework that simplifies the development of Java applications by providing a comprehensive set of features and tools. It builds on the Spring framework and introduces auto-configuration, which reduces the need for extensive manual setup and configuration. Spring Boot’s primary goal is to enable developers to create stand-alone, production-ready applications quickly and with minimal effort.

### Conclusion

Therefore, Angular, ng-zorro, HTML, CSS, and Spring Boot were ideal for the project since they fulfilled the goals of the project as well as enabling a smooth development process. Angular and ng-zorro enabled creating an innovative and engaging front-end with better and more effective components and layout responsiveness. On the other hand, Spring Boot was a strong back-end framework that provided a good base for managing the business logic and execution of data operations where the application was well constructed and scalable for web base applications.

Front-end and back-end integration was done to form a feature rich, smooth, and user-friendly application that would also satisfy performance and user expectations set by the project.

# Chapter 4: Requirements

### Introduction

In the realm of software development, requirements play a vital role in shaping the design and functionality of an application. These requirements can be broadly categorized into two main types: Functional Requirements and Non-Functional Requirements.

### Functional Requirements

* + - Expense Management: Create, Read, Update, Delete (CRUD) Expenses: The users should be able to enter new expenses, display all the expenses, edit details of a certain expense and even delete the expense incase it is not needed any longer.
    - Income Management: Income Categorization: Every income should be classified into sections which include but not limited to product sales, services or any other source of income in order to simplify the financial transactions.
    - Employee Management: Add and Manage Employees: Users should be allowed to create new employees into the newly developed system entering aspects like name, contact details, position as well as salary. Users should also be able to add and edit their details as well as delete an employees record.

.

* + - Dashboard for Financial Overview: Expense and Income Reports: Users should be able to analyze the company’s financial performance through various reports including; Total Expense, Total Income and Expense and Income at the category or employee level.
    - Authentication and Security: Secure Login System: It also needs users’ authentications and I mentioned that they should use the right details to log in the system and get full access to it together with its features. This guarantee that only those personnel who have business with it can handle financial data.
    - Expense and Income Tracking: Real-time Updates: It should facilitate the real time updating of the expenses and the amount of income being incurred or received. When data is entered users should be able to immediately see new changes and follow updates.
    - Notifications and Alerts: Expense Limit Alerts: A recommendation is also that users should be able to get notification or alert when a certain level or expense has been exceeded.

These functional requirements also guarantee that BalanceLab meets all the necessary functional needs of the companies within its domain and offers the best solution to address its financial needs, in addition to providing simple, efficient, and highly-manageable control over the fundamental financial operations.

.

### Non-Functional Requirements

* Performance and Response Time, the system result that is showing user performed action like create, modify, view records of a particular system should take maximum 2 seconds in normal network traffic environment.
* Security and Data Encryption: All the financial and personal details should be encrypted – while transferring and storing them (for instance, login credentials).
* Usability and User Interface: The layout of the application should clearly indicate the various sub sections and should have clear labels of button/ function control. It should be possible to interface with the system by anyone without a lot of training in the system.
* Reliability and Availability: The system has to be available 99% of the time at the very least and uptime is the most critical metric for any online services. Synchronous 9% of the time and asynchronous 1% of the time while users have constant access except when there is maintenance.
* Backup and Recovery: The system must perform a backup copy at least once a day and the possibility to restore the lost information in a time period up to 24 hours.
* Maintainability and Modularity: The code should be modular and conform to standards that are present in the industry to make it easier to maintain and expand.

These non-functional requirements guarantee that BalanceLab not just operates as it should, but also gives a protected, easy and steady method for controlling the company’s monetary situation.

### Conclusion

By incorporating these well-thought-out functional and non-functional requirements, BalanceLab is built to be a practical and reliable solution for managing financial operations. The application simplifies the process of tracking expenses, managing incomes, and handling employee data, all while ensuring security and ease of use. This thoughtful design guarantees that the system is not just technically sound but also intuitive for everyday users. It allows organizations to focus on making better financial decisions with confidence, knowing that the system is flexible, responsive, and built to grow with their needs. Ultimately, BalanceLab is crafted to make financial management easier, more transparent, and more efficient for everyone involved.

# Chapter 5: Graphical Interpretation

### Introduction

In this segment, we will explore the numerous components of our financial management software, BalanceLab, through a series of visual representations. By inspecting the provided screenshots, we can gain a deeper understanding of how each feature contributes to the overall financial tracking process.

Through these graphical interpretations, we aim to showcase the user-friendly interface, core functionalities, and the financial insights that BalanceLab provides. Each image will be accompanied by a detailed explanation, emphasizing its relevance and how it enhances the user experience. Whether managing expenses, tracking incomes, or viewing reports, every part of the software has been designed to streamline financial management for our users.

### Website Abilities

### Registration Page :

A screenshot of a computer

Description automatically generated

*Figure 7: Security detail*

A screenshot of a computer

Description automatically generatedWhen a user navigates to the /sign up page within the BalanceLab gadget, they're required to go into a password earlier than proceeding. This more layer of safety guarantees that best users with the proper get admission to code can sign up an account. Once the password is verified, the user can continue with the registration procedure

*Figure 8: Register User*

Once the password is verified, the user can continue with the registration procedure through entering their private information like username, password, first name, last call, cellphone range, and profile picture. This step allows protect the system from unauthorized registrations*.*

A screenshot of a computer

Description automatically generated

*Figure 9: Login Page*

After registration user need to enter his credentials in order to login to home page .

### Home Pages:

### A screenshot of a computer Description automatically generated

*Figure 10: Home Page*

### Live Charts:

A screenshot of a graph

Description automatically generated

*Figure 11: Income and Expense Charts*

In this charts we can see the incomes and expenses of the company by day, using them would help the accountant to see if the company is financially stable or not.

The BalanceLab dashboard affords customers with an at-a-glance view of their monetary state of affairs. It offers a clean summary of key monetary metrics, which include the overall profits, general fees, and the modern stability. The dashboard also capabilities visual representations of earnings and expenses, assisting users speedy recognize the monetary distribution. Additionally, it consists of a phase for current transactions, supplying short get entry to to the most recent profits and expense entries. The interface is intuitive, permitting customers to without problems navigate among exclusive sections, consisting of charges, incomes, and employee control, making sure efficient financial tracking and management.

### Incomes Page:

A screenshot of a computer

Description automatically generated

*Figure 12: Incomes Page*

In this web page, the consumer interface is split into fundamental sections. On the left, there are a form to publish new profits, where users can input the name, quantity, date, description, and class of the profits. This shape makes it easy for users to feature new profits records to the device.

On the right aspect, the Past Incomes segment lists previously recorded earnings entries. Each access shows the name, amount, date, and outline. There are icons allowing users to both edit or delete the income report, making control easy and efficient.

### Expenses Page:

A screenshot of a computer

Description automatically generated

*Figure 13: Expenses Page*

In this control page, the interface is divided into two primary sections. On the left, there is a shape for posting a new expense. This form permits the person to go into the expense title, quantity, choose the related employee, choose a date, and offer an outline and category for the cost. This established input guarantees that every one’s essential information for coping with fees are accumulated efficaciously.

On the right facet, the Past Expenses phase lists previously recorded charges. Each access suggests vital details which include the price identify, associated worker, amount, date, and class. Users additionally have the option to both edit or delete an access via the pencil and trash icons respectively. This feature lets in users to manage past prices easily, making corrections or removing entries as wished.

A screenshot of a computer

Description automatically generated

*Figure 14: Update Expense*

### Create Employee:

A screenshot of a computer

Description automatically generated

*Figure 15: Create Employee*

The layout presents a clean, simple form for adding new employees to the system. At the top, there is an option to **upload a profile image** Below that, the form includes fields for entering the employee’s **first name**, **last name**, and **email address**,etc…

A screen shot of a computer

Description automatically generated

*Figure 18: Schedule Salaries*

The method “addSalariesAsExpenses()” is a scheduled task that runs at **12 AM on the first day of every month**. It automates the process of adding monthly salary payments as expenses for each employee in the system.

A screenshot of a computer

Description automatically generated

*Figure 20: List of Employees*

This webpage show the list of all the employees with the ability to search for a specific one using the search engine.

### Employee Profile:

A screenshot of a computer

Description automatically generated

*Figure 22: Employee Profile*

After clicking on the view details button we can have access to the employee profile, we can see all employee details including his recent expenses and the abilities to update or delete the employee profile, in top we can see a “+” button that can take u directly to create a new expense for that employee without the need of choosing him manually.

# Chapter 6: Conclusion and Perspective

## GENERAL CONCLUSION

The improvement of BalanceLab marks a tremendous achievement in streamlining economic management for groups, specializing in offering a continuing platform for dealing with costs, earning, and employee facts. This web application has been designed quite simply of use and functionality in mind, permitting users to track and manage their economic information efficiently and correctly.

Throughout the improvement procedure, numerous challenges have been met and resolved, resulting in a robust system that simplifies the once-complicated tasks of financial file-preserving. Features together with automated income costs and actual-time financial insights have verified to be sport-changers, empowering users to preserve a clear view of their financial standing without manual intervention.

The application’s intuitive consumer interface guarantees that even non-technical customers can navigate results easily, whether they may be including new expenses, updating worker statistics, or reviewing economic reports. Moreover, the capability to generate comprehensive financial summaries and track the balance between incomes and costs has greatly more advantageous decision-making techniques inside corporations.

Beyond its technical accomplishments, the creation of BalanceLab represents an adventure of growth, each in terms of software program development and information the importance of green monetary management in today’s rapid-paced commercial enterprise environment.

In conclusion, BalanceLab stands as a powerful device for remodeling economic management. Its success in addressing the wishes of customers, mixed with its capacity for future upgrades, ensures that it's going to preserve to serve as a critical solution for corporations looking to optimize their financial methods.

### Introduction

In the ever-changing global of monetary control, BalanceLab has emerged as a effective device for streamlining and simplifying economic operations. Throughout this report, we’ve delved into the development and implementation of this progressive platform, exploring the vital capabilities that have transformed how companies control fees, earning, and employee statistics.

As we now method the conclusion of our analysis, this introduction sets the level for summarizing the key insights received from our adventure with BalanceLab and reflecting on the profound impact it has had on improving economic transparency, accuracy, and operational performance inside corporations.

### Perfection Constraints

#### Cost classification and prioritization

BalanceLab should enable users to break down costs (e.g., administrative expenses, payroll, utilities) for better financial tracking and planning. The plan should allow for prioritization of necessary funds and ensure that important financial obligations are met immediately. Critical costs should be flagged to keep users aware and avoid significant delays in payment.

#### Real-time financial insights and reporting

BalanceLab should provide users with real-time financial information and insights. This includes comprehensive financial statements such as income statements, cost breakdowns and balance sheets. The system should provide visual data (e.g., charts, graphs) for easier analysis and faster decision-making.

#### Employee cost selection

BalanceLab allows for easy expense management for employees, ensuring that each employee’s financial impact, such as reimbursements or salary payments, is tracked This ensures accurate records are kept and expenses and economic activity of management is well aligned.

#### User access and data security

The application should enforce strict access controls, ensuring that only authorized users have access to sensitive financial information. The system should include role-based authorization to restrict access to specific areas of the platform, protect sensitive information, and ensure compliance with data security standards

## 

## Resources

1. **Angular Definition:** https://angular.dev/overview

**2.CSS Definition:** [**HTTPS://WWW.W3SCHOOLS.COM/CSS/CSS\_INTRO.ASP**](https://www.w3schools.com/css/css_intro.asp)

**3.NG-ZORRO Definition:** https://ng.ant.design/docs/introduce/en

**4.** **JAVA Definition:** <https://www.java.com/fr/download/help/index_using.html>

**5.Spring Definition:** https://spring.io/why-spring

**6.TalkLab History:** https://www.talklab.ma/