Timelink

Final Report

IT in a real organization

André Martins,
Manon Barthelemy,
Elvin Cocco,
Leonardo Coimbra,
Tomás Coimbra

Lamtco Solutions

0230991223@uni.lu

December 13, 2025

Contents

1 Introduction

In this document you will find the a summary of the work that has been accomplished by the whole team for this project.

2 Risk Management

2.1 Risk Register

Risk name	Risk Description	Risk Impact	Solution
Regulatory non-compliance (GDPR)	The system can expose sensible data (name, ID, hire date) without respecting GDPR.	High, legal risk and reputation risk	We can either accept this risk because our project will not be public, or we can try to add compliant measures (hahsing)
Data Privacy	A user see sensible data, but he doesn't have the rights to see these.	High, legal risk and reputation risk	We need to decide which type of user can see what type of informations and then implement these. Or accept this risk.
Incorrect data, not-updated, missing	Th represented data (shifts, working hours, employees assigned to a working place, working place details, etc. . .) are not synchronized correctly because of a bug, a failure,	High	We accept it for the moment.

Risk name	Risk Description	Risk Impact	Solution
Insufficient log retention (for activity log UC-12)	Logs expire too soon or are not archived	Low	Apply the right parameters (choose the right time to keep activity logs)
Modification conflict	Two administrators modify the same data at the same time.Overwriting of modifications, inconsistency between assignments.	High	When developement, think of concurrent modification to prevent this risk
No validation prior to major modification (add, remove, edit a user or a working place)	No confirmation before modification or deletion.Risk of irreversible human error.	High	Implement a validation system, a message (“Are you sure you want to. . . ”)
Poor management of duplicates	Users, project, working places, etc.. are created multiple times	Medium	When developement, think of managing duplicates to prevent this risk
Authentication service unavailable	Admin can no longer access their account, critical operations blocked.	High, service interruption	Accept this risk.
Unautorized access, security failure	A compromised admin account, or a password leak that allow access to the system.	High, GDPR sanctions, data breaches, manipulation of sensitive information	Use security measures (multi-factor authentication, strong password, etc. . .)

2.1.1 QA Test Report

Manual Tests

UC #	Name	Preconditions	Steps	Expected Result	Actual Result
1	Monitor Employee Shift	- User connected to the Internet- User logged into the application	- Open application- Click button “Review work” to go to the review page- Select the time period	- Employee’s working hours are displayed for the selected time- Employee’s information is displayed, within 2 seconds	

UC #	Name	Preconditions	Steps	Expected Result	Actual Result
1	Monitor Employee Shift (no time period selected)	- User connected to the Internet- User logged into the application	- Open application- Click button “Review work” to go to the review page	- Employee’s working hours of the month are displayed - Employee’s information is displayed, within 2 seconds	

UC #	Name	Preconditions	Steps	Expected Result	Actual Result
1	Monitor Employee Shift (no working hours for this time period)	- User connected to the Internet- User logged into the application	- Open application- Click button “Review work” to go to the review page- Select the time period	- Warning message: “No working hours for the selected time period.”	

UC #	Name	Preconditions	Steps	Expected Result	Actual Result
2	Start a Shift	- User connected to the Internet- User logged into the application- Application has GPS allowed	- Open application- Go to the Shift Page- Click button: "Start shift"	- Shift has started within 2 seconds- Timestamps and GPS saved- Message: "Shift started at hh:mm"	

UC #	Name	Preconditions	Steps	Expected Result	Actual Result
2	Start a Shift (offline)	- User not connected to the internet- User logged into the application- Application has GPS allowed	- Open application- Go to the Shift Page- Click button: "Start shift"	- Application saved shift data locally- Message: "Data will be synchronized when you are online"- Response within 2 seconds- Data is sent to the server when online	

UC #	Name	Preconditions	Steps	Expected Result	Actual Result
2	Start a Shift (no GPS)	- User connected to the internet- User logged into the application- GPS not allowed on the App	- Open application- Go to the Shift Page- Click button: "Start shift"	- Shift has started within 2 seconds- Timestamps only is saved and GPS value are null- Data is sent to the server- Message: "Shift started at hh:mm"	

UC #	Name	Preconditions	Steps	Expected Result	Actual Result
3	End a Shift	- User connected to the Internet- User logged into the application- Has an active shift	- Open application- Go to the Shift Page- Click button: “End shift”	- Shift has ended within 2 seconds- Hours saved and total number of hours computed without errors- Message: “Shift ended total: hh:mm”	

UC #	Name	Preconditions	Steps	Expected Result	Actual Result
3	End a Shift (offline)	- User not connected to the Internet- User logged into the application- Has an active shift	- Open application- Go to the Shift Page- Click button: “End shift”	- Shift has ended within 2 seconds- Data stored locally, auto synchronization when online- Message: “Shift ended total: hh:mm”	

UC #	Name	Preconditions	Steps	Expected Result	Actual Result
4	Employee Reminder (Part 1)	- User connected to the Internet- User logged into the application- Notifications for this application are enabled- At least one available shift is associated with the reminder- User has configured a reminder for its shift (example: 5min before the start of the shift)- The specified time for the notification to be send is reached	///	- Notification is send to the user (“Your shift starts in … minutes.” / “Your shift ends in … minutes.”) within 2 seconds.	

UC #	Name	Preconditions	Steps	Expected Result	Actual Result
4	Employee Reminder (Part 2)	- User connected to the Internet- User logged into the application- Notifications for this application are enabled- At least one available shift is associated with the reminder- User has configured a reminder for its shift (example: 5min before the start of the shift)- The notification has been sent	- User opens the notification on its device- User click on its notification	- Applocation opens directly to “Start a shift” / “End shift” screen.	

Unit & Integration Tests

Unit test summary During the execution of the unit tests, several issues were discovered, mostly related to incorrect business logic in the backend layer. For example, the verifyEmail() method returned false when the email did not exist in the system, while logically it should only return false when the email already exists. These issues demonstrate the usefulness of unit tests in detecting logic defects very early in the development phase.

Most modules passed unit tests successfully, including:

- Service layer
- Domain and DTO objects
- Exception handling
- Security configuration

However, the Controller tests encoutered problems and were unable to complete execution.

Overall, unit testing revealed multiple design inconsistency that need to be fixed before release. The test campaign proved highly valuable for improving code reliability, especially on core business logic.

List of executed tests The unit tests cover all critical features of the application: EmployeeServiceTest

- Employee creation (validations, admin role, email conflicts)
- Update
- Deletion
- Retrieval by ID or Email
- Admin access vs. self-identity access
- Authorization checks
- Exceptions on invalid data

ProjectServiceTest

- Project creation
- Project number conflicts
- Client/project association
- Update
- Deletion
- Full project listing
- Admin access

- Exceptions for missing client or project

TimestampEntryServiceTest

- Timestamp creation
- Update
- Deletion
- Global and per-ID retrieval
- Duration/GPS coordinate validation
- Exceptions for missing employee/project or invalid duration

CustomerServiceTest

- Valid customer creation
- Valid/invalid email
- Customer update
- Customer deletion
- Customer search
- Admin validation
- Exceptions based on data compliance

AuthServiceTest

- Successful authentication
- Encrypted password verification
- Token generation
- Non-existing email
- Invalid credentials

UserAuthDataTest

- Creation of authentication users with or without constructor arguments
- Equality between instances
- Proper hashCode() and toString() implementation

AuthRequestTest

- Validation of both constructors
- Getters / setters behavior
- Data consistency

Each entity test includes:

- field initialization
- consistency of bidirectional relationships
- persistence of stored values

Enums Role & Tag

- Validation of enum constants
- Behavior of valueOf()

DTO tests validate:

- creation with or without attributes
- setters behavior
- field updates
- acceptance of null values when allowed

ExceptionsTest

- Verification of business exception hierarchy
- Correct initialization of message and cause

Integration AuthControllerIT

- Login validated via password hash
- Checks for wrong password and non-existent user

CustomerControllerIT

- Retrieval, creation, and database persistence

EmployeeControllerIT

- Full CRUD (read, create, update, delete)

ProjectControllerIT

- Retrieval and creation in client context

TimestampEntryControllerIT

- Creation, reading, updating, and deletion
- Admin vs. self access rules
- Business validation (maximum duration, etc.)

Percentage of tests that passed/failed

Test class	Number of tests executed	**Passed**	**Failed**
Services	45	44	1
Controllers	19	0	0
Domain/Models	10	10	0
DTO	11	11	0
Exceptions	6	6	0
Security	22	22	0
Total	113	113	1

Summary of detected critical errors Only one error was noted, concerning TimestampEntryService.updateTimeStamp: - the test reports a NonConformRequestedDataException related to Timestamp compliance, likely due to validation (duration / start time / business rule). - No other test revealed functional errors. - No errors related to the Repository or dependency injections. - Admin / self-access rules work correctly. - Business constraints (email, existing project, etc.) are properly enforced.

Note: The integration tests for controllers (AuthControllerIT, CustomerControllerIT, EmployeeControllerIT, ProjectControllerIT, TimestampEntryControllerIT) do not complete execution, they remain pending instead of passing or failing.

Conclusion Unit tests demonstrate a stable, consistent, and secure application, where:

- business rules are properly enforced,
- administrative access controls are robust,
- CRUD operations behave correctly,
- exceptions are correctly propagated.

The test coverage is extensive and relevant, covering both normal and exceptional scenarios. The only non-compliance to address concerns the update of a Timestamp, potentially related to overly strict business rule validation or a missing initialization in the mock.

Note: The integration tests for controllers remain pending and do not complete execution, so their actual pass/fail status is not determined.

3 Project Presentation Summary

3.1 Project Status Update

- Successfully delivered the **first version of the working prototype**.
- Core functionalities are implemented and demonstrated during the presentation.

3.2 Project Metrics

- **Milestones:** Key planned milestones were reached as scheduled.
- **Effort Tracking:** Man-days spent were presented to reflect team workload and progress.

3.3 Updated Project Scope

- Implemented:
  - **Mobile application**
  - **Data server / backend**
- Excluded from current scope:
  - **Admin page**, postponed for a later phase to focus on core functionality.

3.4 Testing Overview

Manual Testing

- **Total tests:** 12
- **Passed:** 9
- **Failed:** 3

Unit Testing

- **Total tests:** 113
- **Passed:** 112
- **Failed:** 1

Test Results Summary

- Very high success rate, indicating strong stability of the current implementation.
- Failed tests were identified and documented for future fixes.

3.5 Demonstration

- A live demo was performed, showcasing the main features of the mobile app and its interaction with the data server.
- Demonstration confirmed correct system behavior in realistic usage scenarios.

3.6 Final Conclusion

Project overview

The Timelink project was carried out through a structured development process that covered the full software lifecycle. The main deliverables included:

- Brainstorming and idea definition
- Specification and requirements analysis
- Software architecture design
- UI mockups and blueprint creation
- Back-end development
- Front-end (mobile) development

This approach allowed the team to progress from concept to a functional prototype in a controlled and iterative manner.

Results and highlights

- Delivery of **Timelink Prototype Version 1**, serving as a proof of concept
- A solid and extensible codebase established for future development
- Functional foundations in place for:

  - Security and authentication
  - Back-end data handling
  - Mobile front-end user experience

Challenges and resolutions

Several challenges were encountered during development:

- Part of the initial scope had to be reduced to ensure timely delivery
- Defining a compliant and feasible architecture required multiple iterations
- Team members had to learn and adapt to new programming languages and technologies
- Early design definitions were sometimes lacking, requiring later adjustments

These challenges were addressed through prioritization, continuous discussion, and iterative improvements.

Positive outcomes

Technical outcomes:

- Hands-on experience with Expo and React Native
- Integration and usage of Swagger / OpenAPI for backend communication
- Better understanding of the differences between React Native and React

General outcomes:

- Improved task management and individual responsibility
- Enhanced teamwork and collaboration through coordinated development and testing efforts

---

Lessons learned

- Design phases should be emphasized more and specified in greater detail
- Testing should be introduced earlier in the development process
- Scope creep should be avoided, and contingencies should always be planned for unexpected issues

Overall, the project provided valuable technical and organizational learning experiences while delivering a working prototype of the Timelink system.