

Secure Coding

Phase 1











Team 06

Efdal Ustaoglu

Efe Amadasun

Malte Kessner

Nikolaos Tsiamitros

Use Case	Implemented
User registration (e-mail with TANs)	
User login	
User Logout	
View bank account details of Customer	
View transaction history of Customer	
Customer money transfer via HTML form	
Customer money transfer via uploading transaction file	
Transfer approval for amounts larger than 10.000 EUR	
Approval of registration of Customer or of other employee	
Download transaction history of Customer as PDF	

Live - Demo

Time Tracking

Efdal

Task	Time (hours)
• VM Download & Installation	1
• Planning of the Structure of Database	1
• Designing relational database (Tables, Value Types, Foreign Keys, Views)	2
• Planning of application structure and making a blank pages for a sketch	1.5
• Implementation of db.php file	2
• After Efe's some of the changes in db.php, complete it where he stopped	1
• Download PDF file	1.5
• Sending e-mail function	2
• Creation of Tan numbers	1

Time Tracking

Efdal

Task	Time (hours)
• Sending Tan numbers in e-mail as "Text"	0.5
• Sending Tan numbers in e-mail as "PDF"	0.5
• Solving issues when running in VM	2
• Merging branches and solving the conflictions	2
• Changing the design of transactions pdf	1
• Bug fixing in backend	2
• Bug fixing in frontend	2
• Presentation Introduction	1
• Presentation Use Cases 1-5	1
• Presentation Use Cases 6-10	1
Total = 26 hours	

Time Tracking

Efe

Task	Time (hours)
• Database planning	1.5
• Database creation SQL script (tables creation, foreign keys, indexes)	2
• App structure planning and creating empty starting files (folder structure, 3 private function files and 10 public display files)	1.5
• Write empty function stubs (with parameters) for the functions that will be needed in the app	2
• Setup CSS framework, and add additional custom CSS styling for app	1
• Create layout files (header.php, footer.php)	0.5
• Functions to create & close DB connections, as well as perform an query that returns result set (SELECT) and doesn't (INSERT, UPDATE, DELETE)	2
• Functions to handle user database operations (login, select all, select one, register, etc)	2
• User interface for login and user registration	2

Time Tracking

Efe

Task	Time (hours)
• Bug fixing for various functions written by me and others	2
• User interface for transactions (create new, view all, view single, approve/deny)	2
• Fixing merge conflicts on multiple occasions with other branches	2
• Function to upload transaction file, run it on OS, and capture result	2
• Connecting disparate functions for user registration approval (sending email, creating tans), and also bug fixes	2
• Prepare some part of the presentation document	2
Total = 26.5 hours	

Time Tracking

Malte

Task	Time (hours)
• Initial software setup and familiarization with the task	1.5
• Database design	1.5
• Software architecture design	2
• File parsing – Implementation: General information extraction	2
• File parsing – Implementation: Enhancement of robustness and flexibility	2
• File parsing – Testing and bug fixing	2
• File parsing – Documentation / Commenting	1
• TAN generation in C (not used in final version)	1
• Email sending in C (not used in final version)	1.5

Time Tracking

Malte

Task	Time (hours)
• DB operations with C – Familiarization with the topic	2
• DB operations with C – Implementation: Basic money transfer	2
• DB operations with C – Implementation: TAN verification, Update of TAN status, transaction history etc.	2
• DB operations with C – Implementation: Enhancement of robustness and cleanup of code	1
• DB operations with C – Testing and bug fixing	1
• DB operations with C – Documentation / Commenting	0.5
• C interface explanation and testing	2
Total = 25 hours	

Time Tracking

Nikos

Task	Time (hours)
• Insert transaction - Validate input data format before inserting transaction to the database	1
• Insert transaction - Check if user input data correspond to valid database records	1.5
• Insert transaction - Create necessary database functions to retrieve data to compare with user input data	2
• Update transaction - Create necessary database functions to access relevant records	1.5
• Update transaction - Check if transfer is possible, based on account balance	1.5
• Update transaction - Update account balance to complete the transfer	1.5
• View transactions page - Create function to present data from database in user friendly format	1.75
• View transactions page - Send the required data to user	0.5
• View transaction page - Create functions to present data in user friendly format	1.75

Time Tracking

Nikos

Task	Time (hours)
• View transaction page - Sending required data to user	0.5
• Creating Generate PDF function	1.5
• The 3 Presentation slides	1
• Time tracking slides	2
• User Guide: First 6 use case slides	2
• User Guide: Remaining 4 use case slides	1
• Set up VM, Download, install, read material	1
• Database Design – Define tables, fields, keys, relations	1.5
• Design the application layers – app/, public/ - with rest of team, help to define main application functions for each level to provide functionality	2

Time Tracking

Nikos

Task	Time (hours)
• Help Efe to resolve conflicts while merging my branch to master	1.5
• Help team to re-design / re-implement functionalities initially implemented on my branch	2
Total = 29 hours	

Use Cases

Employee user of the web application: username: amadasun@in.tum.de
password: pass

Name	Registration
Goal	Registration of a new user (Client/Employee)
Actors	Client / Employee
Pre-conditions	User must not be already registered.
Main Course of Execution	User must fill in his/her first name, last name, e-mail, user type, and a password twice.
Alternate Courses	-
Exceptions	User already registered, Malformed e-mail address, passwords do not match
Post-conditions	User must be approved by an employee
Data formats used	-

Use Cases

Name	Login
Goal	Login to an (approved) account
Actors	Client / Employee
Pre-conditions	User must have registered and his registration must have been approved
Main Course of Execution	User must enter a valid e-mail and password.
Alternate Courses	-
Exceptions	Invalid e-mail / password (user either not registered or not approved)
Post-conditions	User can access pages that his user type authorizes him to.
Data formats used	-

Use Cases

Name	Logout
Goal	Logout from an account
Actors	Client / Employee
Pre-conditions	User must be logged in
Main Course of Execution	User clicks on “Logout” button
Alternate Courses	-
Exceptions	-
Post-conditions	User is not authenticated to view pages that require authentication User is redirected to the login page
Data formats used	-

Use Cases

Name	View Customer Details
Goal	View account details for a specific customer.
Actors	Customer / Employee
Pre-conditions	<ul style="list-style-type: none">• Client is authorized to view only his account details• Employee can view details of all accounts
Main Course of Execution	<ul style="list-style-type: none">• Client clicks on the “User” button• Employee clicks on the “User” button and then picks an account
Alternate Courses	-
Exceptions	-
Post-conditions	User is presented with the account details
Data formats used	-

Use Cases

Name	View Customer Transaction History
Goal	View transaction details for a specific customer.
Actors	Customer / Employee
Pre-conditions	<ul style="list-style-type: none">• Client is authorized to view only his account details• Employee can view details of all accounts
Main Course of Execution	<ul style="list-style-type: none">• Client clicks on the “Transaction” button.• Employee clicks on the “Transaction” button and then picks an account.
Alternate Courses	-
Exceptions	-
Post-conditions	User is presented with the account details
Data formats used	-

Use Cases

Name	Money Transfer via HTML Form
Goal	Transfer money to a different account
Actors	Customer
Pre-conditions	<ul style="list-style-type: none">• Customer must be logged in.• Customer account must have enough balance.• Recipient account must exist in the bank.• Customer should have a valid TAN number.
Main Course of Execution	Customer must fill in a form with the recipient account, amount of money and a TAN number.
Alternate Courses	Customer can upload a text file.
Exceptions	<ul style="list-style-type: none">• Not enough money• Invalid recipient account• Invalid TAN number
Post-conditions	<ul style="list-style-type: none">• If amount is greater than 10.000 EUR it should be approved by an employee. Otherwise it is immediately approved.
Data formats used	text

Use Cases

Name	Money Transfer by Uploading File
Goal	Transfer money to a different account
Actors	Customer
Pre-conditions	<ul style="list-style-type: none">• Customer must be logged in.• Customer account must have enough balance.• Recipient account must exist in the bank.• Customer should have a valid TAN number.
Main Course of Execution	Customer must upload a file containing the sender account, recipient account, amount of money and a TAN number.
Alternate Courses	Customer can fill in an HTML form.
Exceptions	<ul style="list-style-type: none">• Not enough money• Invalid recipient account• Invalid TAN number
Post-conditions	<ul style="list-style-type: none">• If amount is greater than 10.000 EUR it should be approved by an employee. Otherwise it is immediately approved.
Data formats used	Text file

Use Cases

Name	Transfer Approval
Goal	Approval / Denial of a pending transaction
Actors	Employee
Pre-conditions	<ul style="list-style-type: none">• Transaction amount must be greater than 10.000 EUR.• Employee must be logged in.• Transaction state must be "Pending".
Main Course of Execution	Employee clicks on a specific pending transaction and then clicks on "Approve" or "Deny".
Alternate Courses	-
Exceptions	<ul style="list-style-type: none">• The account has not enough balance.• Recipient account does not exist.
Post-conditions	The amount is transferred to the specified account.
Data formats used	-

Use Cases

Name	Registration Approval
Goal	Approval / Denial of a pending user registration
Actors	Employee
Pre-conditions	<ul style="list-style-type: none">• Employee must be logged in.• Registration state must be “Pending”.
Main Course of Execution	Employee clicks on the “Open” button for a specific user and then on “Approve” / “Deny”.
Alternate Courses	-
Exceptions	-
Post-conditions	If the user is approved he/she can log in.
Data formats used	-

Use Cases

Name	Transaction History Download
Goal	Download a PDF containing the transaction history of a specific customer.
Actors	Customer / Employee
Pre-conditions	<ul style="list-style-type: none">• User must be logged in.• Client is authorized to download only his transaction history.• Employee can download transaction history of all accounts.
Main Course of Execution	User clicks on “Download Transactions” button.
Alternate Courses	-
Exceptions	User does not have any transaction history.
Post-conditions	A PDF is downloaded.
Data formats used	-