

# Team 9

~ Alexis Engelke, Johannes Fischer, Ralph Schaumann, Saurabh Nawalgaria ~

Presents:

The SecureBank

# Use cases

- Customer / employee registration
  - Registration has to be approved, TANs sent on approval
- Login and Logout
- Customer / employee can view account details
- Customer / employee can view transaction history and export this as PDF
- Customer money transfer via HTML form or uploaded transaction file
  - Transactions > 10,000 € have to be approved
  - Batch processing supported

# Time Table

- Alexis Engelke (22.5 h)

- Basic Site Structure: 2 hours
- Database driver: 2 hours
- Database tables: 1.5 hour
- Login: 2 hours
- Design updates: 1 hour
- Transaction UI: 1 hour
- Transaction approval UI: 0.5 hours
- MySQL-C-API: 2 hours
- Binding C parser to PHP application: 2 hours
- Improving C parser: 2 hours
- Batch processing: 2 hours
- TAN generation: 2 hours
- Registration approval UI: 1 hour
- Fixing PDF export for employees: 0.5 hours
- Testing: 1 hour

# Time Table

- Johannes Fischer (28h)

- PHP Page Builder Prototype (2h)
- Build Test HTML partials (1h)
- Test HTML partials with Page Builder Prototype (1h)
- Added HTML partials for basic functionality (2h)
- Changed all of index.php to use HTML partials (2h)
- Added partials for the user's home (1h)
- Testing several use cases (2h)
- Debugging (2h)
- PDF export create dummy file (2h)
- PDF export implementation with dummy values (2h)
- PDF export implementation finish (real values) (2h)
- Fixed errors of PDF creation (2h)
- Fixed users' home site (2h)
- Fixed UI problems (showing more info now) (2h)
- Fixed transaction approval (2h)
- Help with mail (1h)

# Time Table

- **Ralph Schaumann**

- Setting up an appropriate C environment (CLion freezes) 2
- Creating the file formatting and layout 1
- Programming the first basic parser 2
- C Parser optimizations, error correcting 2
- Final testing of the parser 1
- Authentication with dual password, implementation of max 3 attempts 2
- Testing and fixes 1
- HTML/CSS layout 2
- Looking up (my own) old code for mail sender 1
- Translating java mail sender to php 1
- Testing and fixes (php = pretty horrible preprocessor-language) 2
- Implementation of a Pseudo-Captcha 1
- General code testing and notes for fixes 2
- Mail Sender solution search 2
- Mail Sender integration 2
- Mail Sender Testing 1
- Installation/creation of a debian (client) freebsd (host) jail 2
- Installation of software packages for SMTP server 2
- creating smtp relay 1
- certificate creation for TLS 2
- Mail Sender reprogramming to use TLS 2
- certificate creation, installation, transfer, etc 1
- Securing SMTP Server back-end 1
- Testing SMTP Server security 2

—

# Time Table

- Saurabh Nawalgaria

- Implementing Transaction Performing in PHP 2
- Worked on Generating Tan Numbers in C 2
- UseCases Documentation(Register Login Logout View Account details) 2
- UseCases Documentation(PDF Export, Transaction via HTML and via File upload) 2
- Misc(Fixing Issues and code improvement) 2

—

# Registration

Goal	End-user (new-customer or employee) wants to get an account at the SecureBank.
Actors	Customer/Employee
Pre-Conditions	End-user found the web-page
Main Course of Execution	The end-user enters his name, e-mail and password and clicks "register"
Alternate courses	None
Exceptions	If the e-mail address is not unique or the passwords don't match, an error is shown
Post-conditions	An employee gets a notice about the registration. After the approval of an employee, an e-mail with TANs gets sent and the user can use its login as well as perform transactions.
Data-formats used	Text / E-Mail

# Login

Goal	End-user (customer or employee) wants to log into his account
Actors	Customer/Employee
Pre-Conditions	End-user registered
Main Course of Execution	The end-user enters his e-mail and password and clicks “login”
Alternate courses	None
Exceptions	If the e-mail address is not known or the password is wrong, an error message is shown
Post-conditions	The user is logged in and can perform transactions as well as see his transaction history
Data-formats used	Text / E-Mail



# Logout

Goal	End-user (customer or employee) can log out of his account
Actors	Customer/Employee
Pre-Conditions	End-user is logged in
Main Course of Execution	The end-user clicks “logout”
Alternate courses	None
Exceptions	None
Post-conditions	The user is no longer logged in.
Data-formats used	None

# Bank account details of customer

Goal	End-user (customer or employee) can view the account details, such as the account balance
Actors	Customer/Employee
Pre-Conditions	End-user is logged in and knows the account ID
Main Course of Execution	A customer sees the information on his homepage, an employee can enter the ID of a customer to view the information
Alternate courses	None
Exceptions	If the specified account ID doesn't exist, an error message is shown.
Post-conditions	None
Data-formats used	Number

# Transaction history of customer

Goal	End-user (customer or employee) can view the transaction history.
Actors	Customer/Employee
Pre-Conditions	End-user is logged in and knows the account ID
Main Course of Execution	A customer sees the information on his homepage, an employee can enter the ID of a customer to view the information
Alternate courses	None
Exceptions	If the specified account ID doesn't exist, an error message is shown.
Post-conditions	None
Data-formats used	Number

# Money transfer via HTML form

Goal	Customer can transfer money from his account to any other account.
Actors	Customer/Employee
Pre-Conditions	End-user is logged in, knows the account ID of the recipient and knows at least one unused TAN.
Main Course of Execution	The customer clicks on the “New transaction” link on the homepage and enters the required information (recipient, amount in euros, TAN, subject) into the form and clicks on the “Perform Transaction” button
Alternate courses	Money transfer via batch file, see next page
Exceptions	If the specified account ID doesn't exist, the TAN is already used or not existant or the amount cannot be parsed, an error message is shown.
Post-conditions	The transfer happens if the amount is < 10,000 €. Otherwise, an employee has to approve the transfer.
Data-formats used	TAN ([a-zA-Z0-9+/{15}), number (euros and account ID)

# Money transfer via batch file

Goal	Customer can perform multiple transactions with uploading a batch file
Actors	Customer/Employee
Pre-Conditions	End-user is logged in and prepared a batch file. (Format: see next page)
Main Course of Execution	The customer clicks on the “New transaction” link on the homepage, and then clicks on “upload a transaction file”. He uploads the file and clicks “submit”.
Alternate courses	See previous slide
Exceptions	If the specified account ID doesn't exist, the TAN is already used or not existant or the amount cannot be parsed, an error message is shown and all transactions in the file coming after the erroneous transaction won't be executed.
Post-conditions	Each transfer happens if the amount is $< 10,000$ €. Otherwise, an employee has to approve the transfer.
Data-formats used	Transaction Batch format, see next page

# Transaction Batch Format

TOACCOUNT:7

AMOUNT:10.50

SUBJECT:Test Transaction about 10.50 Euros

TAN:abcdefghij12345

----

TOACCOUNT:5

AMOUNT:10000

SUBJECT:Huge transaction about 10000.00 Euros that requires verification

TAN:0123456789abcde

# Approval of large transfers

Goal	End-user (employee) can approve transfers $\geq 10,000$ €
Actors	Employee
Pre-Conditions	End-user is logged in
Main Course of Execution	The employee sees a list of unverified transactions. He can either “Verify” the transaction, so that it gets performed, or “Drop” the transaction.
Alternate courses	None
Exceptions	None
Post-conditions	If the employee clicked “verify”, the transaction gets performed, if the employee clicked “drop”, the transaction gets dropped.
Data-formats used	None

# Approval of registrations

Goal	End-user (employee) can approve registrations
Actors	Employee
Pre-Conditions	End-user is logged in
Main Course of Execution	The employee sees a list of unverified transactions. He can either “Verify” the transaction, so that it gets performed, or “Drop” the transaction.
Alternate courses	None
Exceptions	None
Post-conditions	If the employee clicked “verify”, the new user will be allowed to login. If it is a new customer (no employee), a list of TANs is generated and sent per e-mail. If the employee clicked “drop” instead, the registration gets dropped.
Data-formats used	None



# Transaction history as PDF

Goal	End-user (customer or employee) can export the transaction history as PDF.
Actors	Customer/Employee
Pre-Conditions	End-user is logged in and knows the account ID
Main Course of Execution	A customer clicks on the “Print PDF” link, an employee can enter the ID of a customer and then click on the “Print PDF” link.
Alternate courses	None
Exceptions	If the specified account ID doesn't exist, an error message is shown.
Post-conditions	A download of the PDF file is initiated
Data-formats used	None

# Web application access

- One employee account:
  - E-Mail: `admin@localhost`
  - Password: “superadmin”