# AVOCADO TOAST: AN ONLINE BANKING PLATFORM

DAMH PHAM ANANDITA DUBEY FLAVIU TAMAS
CARLOS DELEON ALEX PETROS

## CONTENTS

1. Planning Scheduling and Peer Evaluation 2
2. Communication and Collaboration 2
   2.1. Screenshot 2
3. Revise and Refine your System 3
4. System Modeling 7
   4.1. Architecture modeling 7
      4.1.1. Logical View 7
      4.1.2. Process View 7
      4.1.3. Development View 7
      4.1.4. Physical View 7
   4.2. Behavioral modeling 7
5. Implementation 8
6. Testing 8
   6.1. Creating Testing environment and Test Cases 8
   6.2. Implementing the Test Cases 13
   6.3. Test Documentation 15

Appendix: Page 16

## 1. Planning Scheduling and Peer Evaluation

| Assignee | Email | Task | Duration | Depends on | Due date | Evaluation |
|---|---|---|---|---|---|---|
| Anandita | Adubey2 | System modeling | 2h | Use cases Class Diagrams | 03/15/19 | 100% completed the task |
| Alex | Apetros1 | Revise and refine your system | 2h | System requirements System modeling | 03/15/19 | 100% completed the task |
| Carlos | Cdeleon1 | Planning Scheduling Communication Collaboration | 2h | None | 03/15/19 | 100% completed the task |
| Damh | Dpham16 | Testing | 2h | Implementation | 03/15/19 | 100% completed the task |
| Flaviu | Ftamas1 | Implementation | 2h | Uses cases Previous imp. | 03/15/19 | 100% completed the task |

## 2. Communication and Collaboration

See Fig. 1

## 3. Revise and Refine your System

### 3.1. ***What is your product, on a high level?***

- Our product is an online banking application, which will allow the customers of a bank or other another financial institution to perform money-management without physically visiting the bank.
- It will allow the customers to open their accounts, manage them electronically, to monitor them, to make transactions, pay their bills, transfer money, make deposits, and so on.

### 3.2. ***Whom is it for?***

- An online banking application is beneficial to everyone but is especially useful for those people with a stringent work schedule.
- It will help them to manage their accounts and keep track of their activities in a quick manner with minimal costs without the need to visit a physical bank during working hours or make a phone call.

### 3.3. ***What problem does it solve?***

- 24/7 availability, saving the customer from rushing the banks during working hours. With an online banking system, people can perform their tasks at a time that suits their work schedule.

- Stringent schedules, where customers with strict working hours to perform their banking activities effectively and conveniently.

- Centralized source of information, where instead of visiting different officials specialized in different tasks, the customer can use an online banking application flexible single enough to do any task in a click. Human bankers are not always available, but the application always will be, meaning there is no need to rush to the bank to get things done.

- Remain informed: With the online banking system, customers can easily receive up-to-date information regarding their upcoming deadlines or dues through notifications, emails, or text messages.

- Easy bill payments, so that there is no need to rush to the bank. Everything can be done at home instead.

## 3.4. *What alternatives are available?*

- Services provided by 3rd party

  application, like Mint A phone app

- In person banking

  services

- Phone calls

- ATMs

## 3.5. *Why is this project compelling and worth developing?*

- It would reduce costs for banks by reducing the amount of human labor re-quired.

- An electronic banking system would enable banks to keep stringent records

- A computerized ledger would reduce the number of mistakes when calculating interests, making transactions, and so on.

- It would increase customer satisfaction because they would have access to our banks from any place with Wi-Fi.

- Providing flexibility to the customers to get their work done at minimal or no cost.

- Saving time of the customers (customer need not necessarily visit the bank physically)

## 3.6. ***Describe the top-level objectives, differentiators, target customers, and scope of your product.***

- Objective: To create a product that performs to our requirements and have it sustainable for multiple lifecycles.

- Differentiators: Our system won't sell data to any third parties, have no hidden transaction fees, and will not participate in predatory lending practices

- Target customers: Our target customer is the average citizen, anyone that currently uses or will use a bank for their transaction needs. (Basically useful to everyone, no matter what)

- Scope: The scope of our project is building an online business ledger, making a database to store our information, developing a web app and making a UI for our customers.

## 3.7. ***What are the competitors and what is novel in your approach?***

- Our competitors are Finacle, nCino, Oracle, etc. What we bring new to the table is that our system will be entirely online.
- Our platform is customer focused and will be fully transparent. We won't sell customer data or charge hidden fees. Additionally, providing the customers with the flexibility to perform the tasks at the minimal or no cost.

## 3.8. ***Make it clear that the system can be built, making good use of the available resources and technology.***

- Our system is possible because it will be very simplistic and direct. Clients will connect to our application software where they will be greeted by a user-friendly interface, which will primarily be coded in HTML, with CSS used to create a beautiful design for our clients.
- Finally, the frontend of our application will be connected to a backend SQL database. The database will be responsible for recording transactions such as withdrawals, deposits, bill pays, and more.

## 3.9. ***What is interesting about this project from a technical point of view?***

The project will use a client-server architecture in order to be accessible to clients from anywhere. Emphasis will also be placed on graphical design, using our creativity to design a nice interface for our clients. Providing flexibility to the clients to perform the non-transactional tasks through online banking. That might include:

- Viewing account
  balances
- Viewing recent
  transactions
- Downloading bank statements
- Downloading periodic account statements
- Funds transfer between the customer's
  linked accounts Paying third parties,
  including bill payments
- Credit card
  applications
- Register utility
  billers

## 4. System Modeling

### 4.1. Architecture Modeling

#### 4.1.1. Logical View See Fig. 2

#### 4.1.2. Process View See fig. 3

#### 4.1.3. Development View See Fig. 4

#### 4.1.4. Physical View See Fig. 5

### 4.2. Withdrawing Cash See fig. 6

### 4.3. Depositing Cash See Fig. 7

## 5. Implementation

5.1. For the backend, open up a console in ‘backend’ and run ‘gradlew.bat run’

5.2. For the front end open up a console in frontend and run ‘npm install; npm start’

5.3. The actual site is [https://avocado-toast.wp6.pw/](https://avocado-toast.wp6.pw/)

5.4. The username is admin and password is admin

## 6. Testing

***1 Testing Environment and Test Cases:***

<u>Environment Test Framework:</u>

In this framework we decided to use:

- NodeJS for Javascript (frontend)
- SQLiteStudio for SQLite (backend)
- IntelliJ IDEA for Java (unit test)

<u>A: Installation of Framework:</u>

*NodeJS:*

Step 1: Go to their website and click download current version for your chosen OS

See Fig. 8

Step 2: Follow the setup wizard form the downloaded file and accept prompts that pop up

See Fig. 9,10,11,12

Step 3: Click install and now you can use the IDE for JavaScript

See Fig. 13

*SQLite:*

Step 1: Go to their website and click download current version for your chosen OS

See fig.14

Step 2: Follow the setup wizard form the downloaded file and accept prompts that pop up

Step 3: Click install and now you can use the IDE for SQLite

See Fig.18

<u>IntelliJ IDEA:</u>

Step 1: Go to their website and click download current version for your chosen OS

See Fig.19

Step 2: Follow the setup wizard form the downloaded file and accept prompts that pop up

See Fig. 20-22

Step 3: Click install and now you can use the IDE for Java

See Fig.23

B: Functionality Testing:

***Identify Features (Functionality):***

1. Login
2. Send Money

***Feature One, Login:***

Partition Input**:** login(String U, string P)

- String U is a string for the user's Username
    - One possible partition is string with length < 0, string with length = 0, string with length > 0
- String P is a string for the user's Password
    - One possible partition is string with length < 0, string with length = 0, string with length > 0

Test Specification:

- Username: <String a-z, A-Z, 0-9, special characters>
- Password: <String a-z, A-Z, 0-9, special characters>

Test Case:

*#1:*

- Inputs:
    - Username: admin
    - Password: admin
- Outputs:
    - Logs in

#2:

- Inputs:
    - Username:
    - Password:
- Outputs:

- Please input your username! Please input your Password!

***Feature Two, Send Money:***

Partition Input: sendMoney(String ID, int N)

- String ID is a string for the recipient's Username
  - One possible partition is string with length < 0, string with length = 0, string with length > 0
- double N is an integer between 0 and the user's maximum balance
  - One possible partition is a number: < -**∞ –** 0.00,0.00-**∞**, 0.00-…*>, …* is the user's maximum balance

Test Specification:

- Recipient's ID: <String a-Z, A-Z, 0-9, special characters>
- Amount: <0…*>, * is the user' maximum balance

Test Case:

Assume user has $10 in balance

*#1:*

- Inputs:
  - ID: bobross
  - Amount: $3.01
- Outputs:
  - Successfully sent $3.01 to bobross

#2:

- Inputs:
  - ID:
  - Amount: $3.01
- Outputs:
  - Please add a destination

*#3:*

- Inputs:

    - ID: bobross
    - Amount:$ -3.01
- Outputs:
    - You can only transfer money out of your account

*#4:*

- Inputs:
    - ID: bobross
    - Amount: $500
- Outputs:
    - You can’t transfer more money than you have

## <u>**2 Implementing the Test Cases:**</u>

<u>Login Functionality:</u>

| | |
|---|---|
| Test ID | UserApiControllerIntegrationTest{} |
| Purpose of Test | This unit test is designed for the software to test our different input partitions for possible strings for username and password to receive the software's output, this way we can see if certain special characters from our test specification would create a bug in the system |
| Test Environment | The environment used is IntelliJ IDE because we wrote our unit tests in Java platform with Windows OS |
| Test Steps | Tester must import the code from our github directory : backend/src/test/java/pw/wp6/avocado_toast/api/UserApiControllerIntegrationTest.java into a Java IDE to proceed running the test |
| Test Input | 1: username: admin password: admin<br><br>2: username: password:<br><br>3: username: kasgponaw password: kasdio |
| Expected Result | 1: user is logged in<br><br>2: String prompt: Please input your username! Please input your password!<br><br>3: User is denied entry |
| Likely Problems/Bugs Revealed | Vulnerability to SQL injections in username and password field to bypass user authentication<br>Vulnerability to SQL injections in username and password field to data manipulation (inserting, deleting, changing data, etc) |

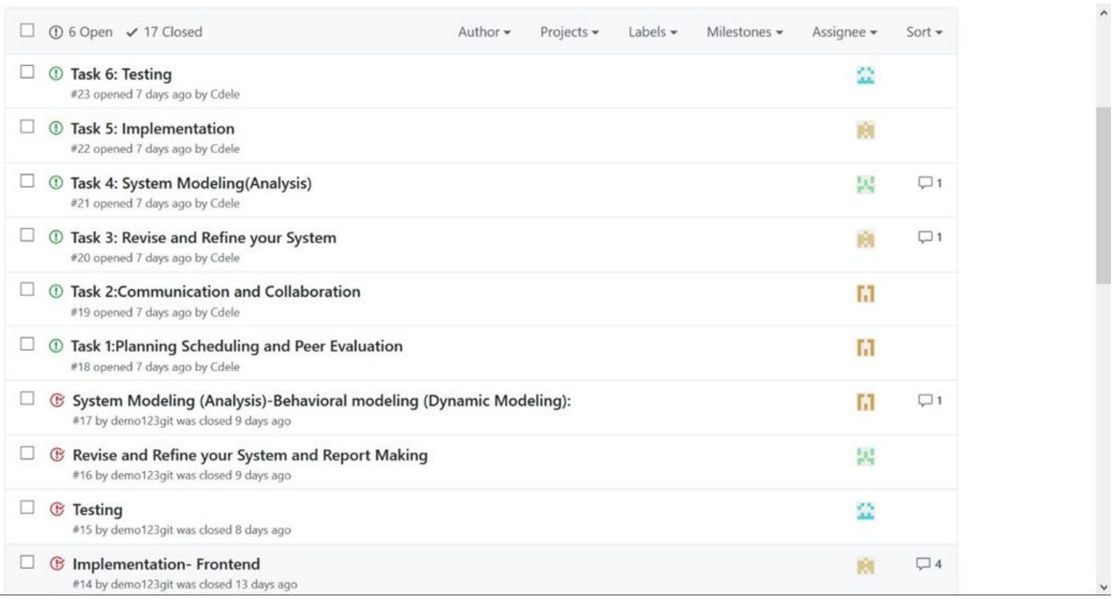
Send Money Functionality:

| | |
|---|---|
| Test ID | TransactionInput{} |
| Purpose of Test | This unit test is designed for the software to test our different input partitions for possible strings for recipient ID and doubles for transaction amount to receive the software's output, this way we can see if certain recipient IDs or transaction amount could bug the system |
| Test Environment | The environment used is IntelliJ IDE because we wrote our unit tests in Java platform with Windows OS |
| Test Steps | Tester must import the code from our github directory:<br>backend/src/test/java/pw/wp6/avocado_toast/api/ LedgerApiControllerIntegrationTest.java into a Java IDE to proceed running the test |
| Test Input | 1: ID: caradelvignee Amount: $10.10<br>2: 1: ID:  Amount: $9.30<br>3: 1: ID: caradelvignee Amount: $-5.20<br>4: 1: ID: caradelvignee Amount: $500.00 |
| Expected Result | 1: Successfully sent $10.10 to caradelvignee<br>2: Please add a destination<br>3: You can only transfer money out of your account<br>4: You can't transfer more money than you have |
| Likely Problems/Bugs Revealed | Vulnerability to SQL injections in recipient ID field to data manipulation (inserting, deleting, changing data, etc)<br>User could enter a negative double for amount and an amount exceeding their balance |

## 3 Test Documentation:

| Bug | The test uncovered the bug | Description of the bug | Action was taken to fix the bug |
|---|---|---|---|
| Vulnerability to SQL injections to bypass user authentication | Login | A user could bypass our authentication software and enter the website without a valid username and password | Change the permission and privileges so user does not have privilege to login if its invalid |
| Vulnerability to SQL injections in username, password, and recipient ID field to data manipulation (inserting, deleting, changing data, etc) | Login<br>Send Money | A user could bypass manipulate our data and insert, delete, or update the data in the username, password, and recipient ID field | Change permission and privilege so user does not have privilege to change data |
| User could enter a negative double for amount and an amount exceeding their balance | Send Money | A user could enter a negative double or an amount they do not have, and this would mess up our data tables with users having an amount they should not have | We added error prompts if the user would enter a negative number or an amount exceeding their balance, and they would be prompted to change the value |

## Appendix:



6 Open ✓ 17 Closed

Task 6: Testing
#23 opened 7 days ago by Cdele

Task 5: Implementation
#22 opened 7 days ago by Cdele

Task 4: System Modeling(Analysis)
#21 opened 7 days ago by Cdele

Task 3: Revise and Refine your System
#20 opened 7 days ago by Cdele

Task 2:Communication and Collaboration
#19 opened 7 days ago by Cdele

Task 1:Planning Scheduling and Peer Evaluation
#18 opened 7 days ago by Cdele

System Modeling (Analysis)-Behavioral modeling (Dynamic Modeling):
#17 by demo123git was closed 9 days ago

Revise and Refine your System and Report Making
#16 by demo123git was closed 9 days ago

Testing
#15 by demo123git was closed 8 days ago

Implementation- Frontend
#14 by demo123git was closed 13 days ago

Fig. 1 List of tasks Complete and in progress

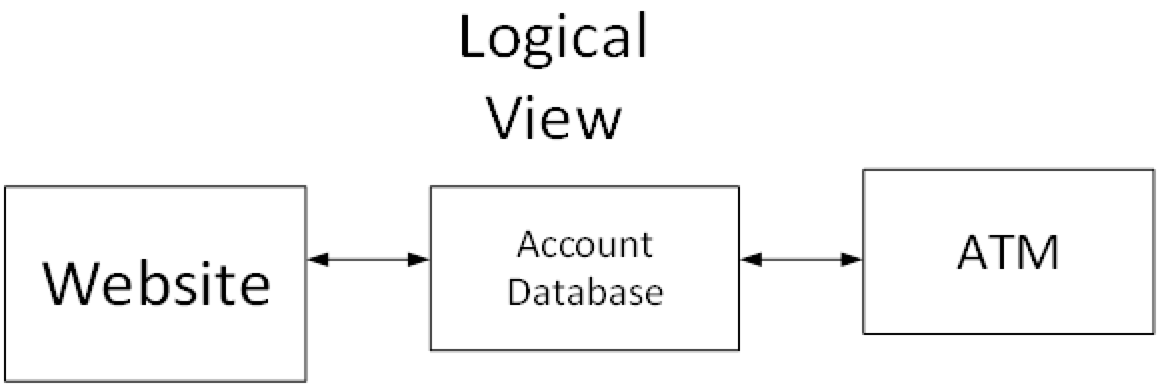


Fig. 2 Logical View

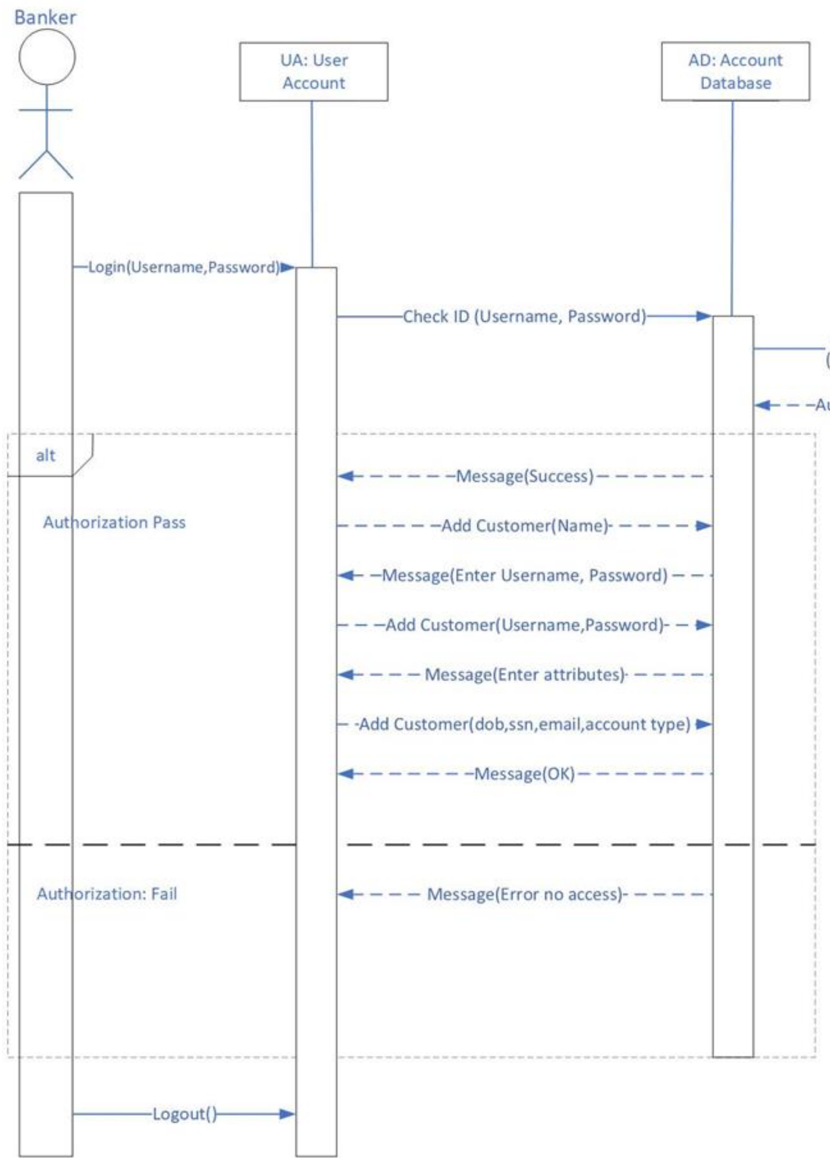


Fig. 3 Process View

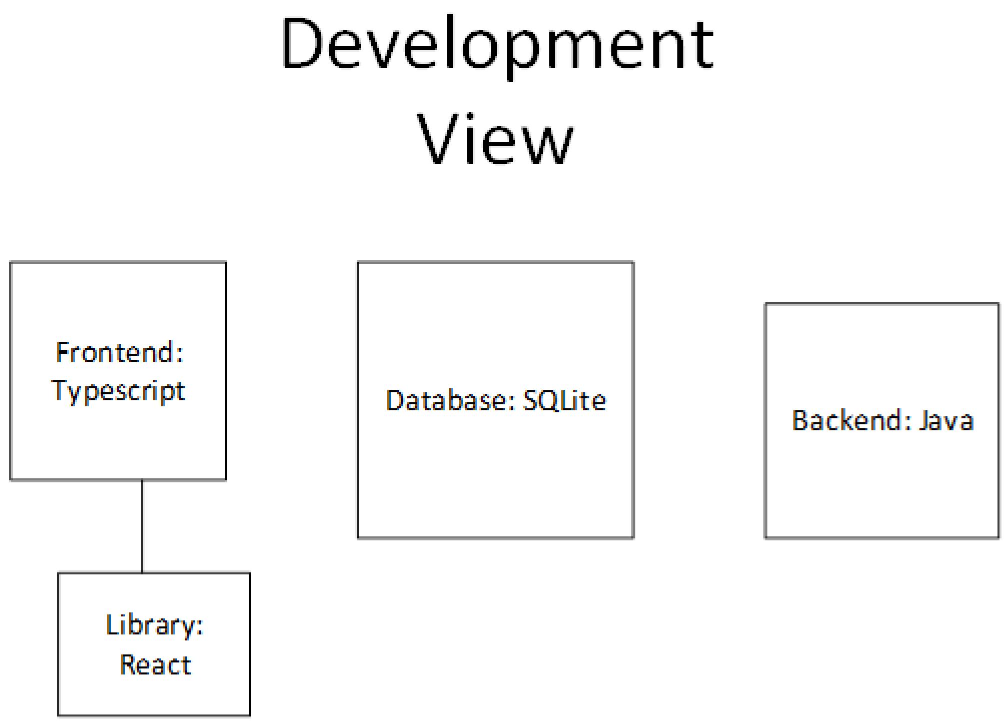


Fig. 4 Development View

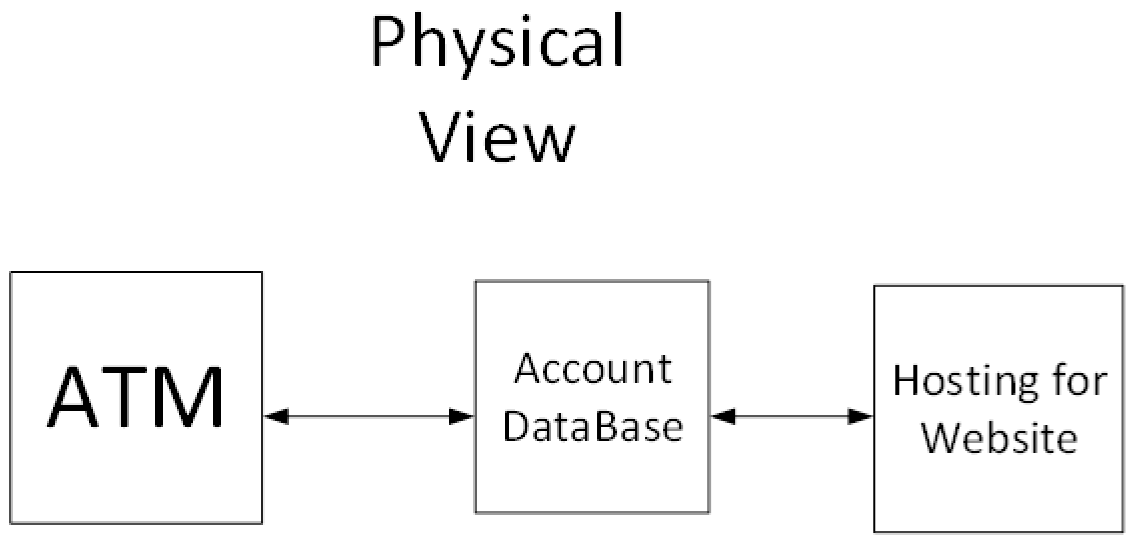


Fig. 5 Physical View

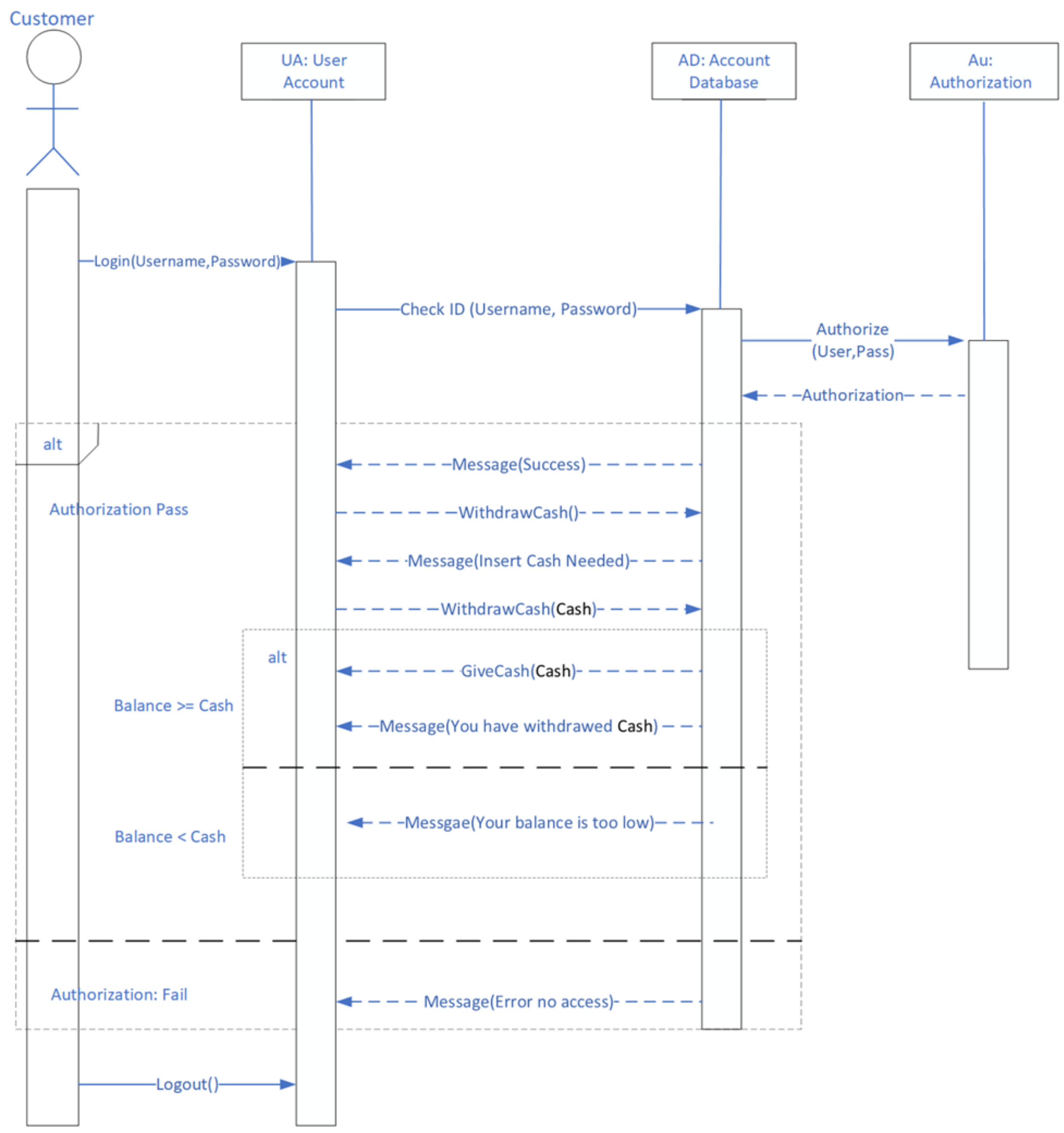


Fig. 6 Withdrawing Cash Sequence

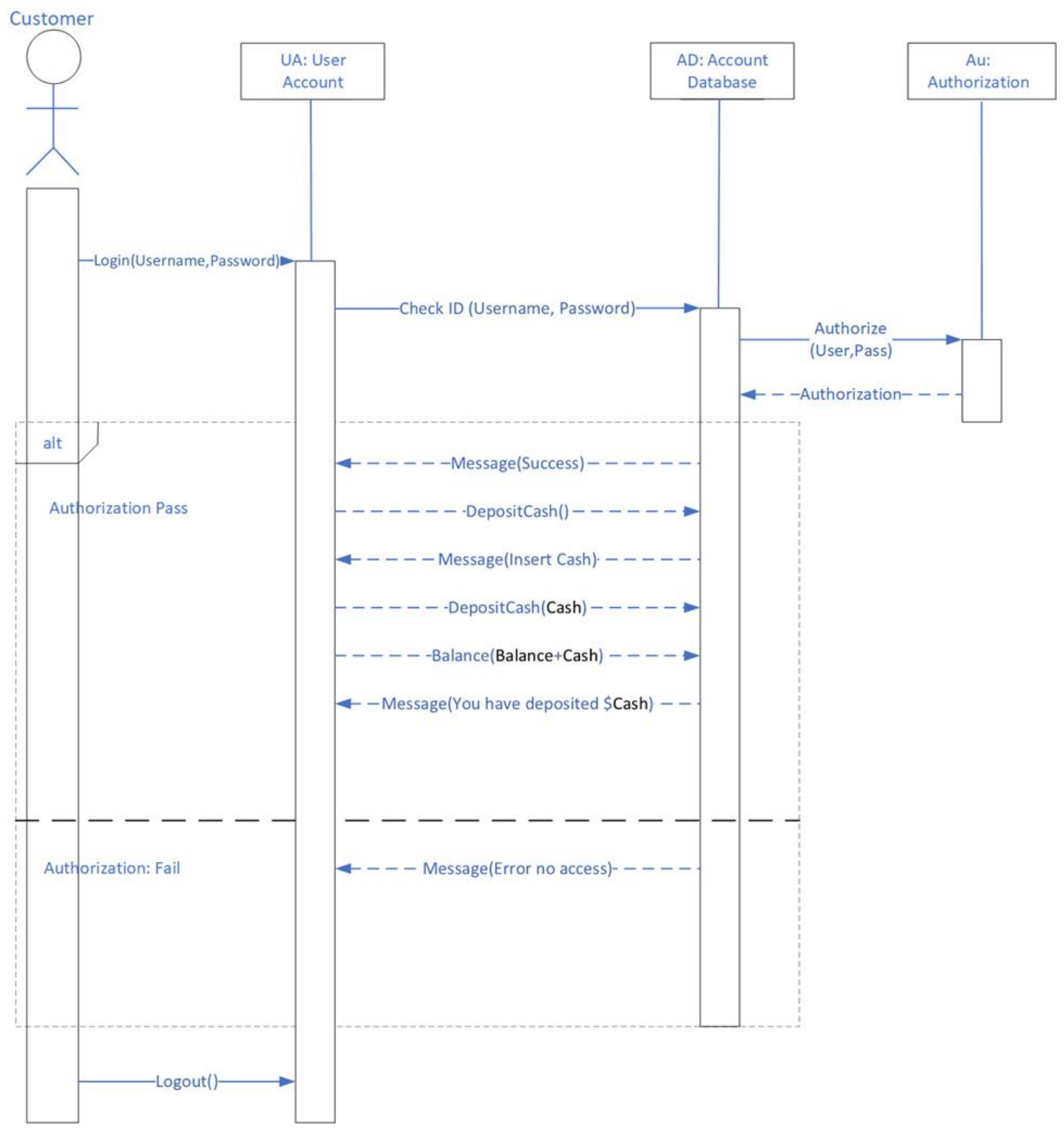
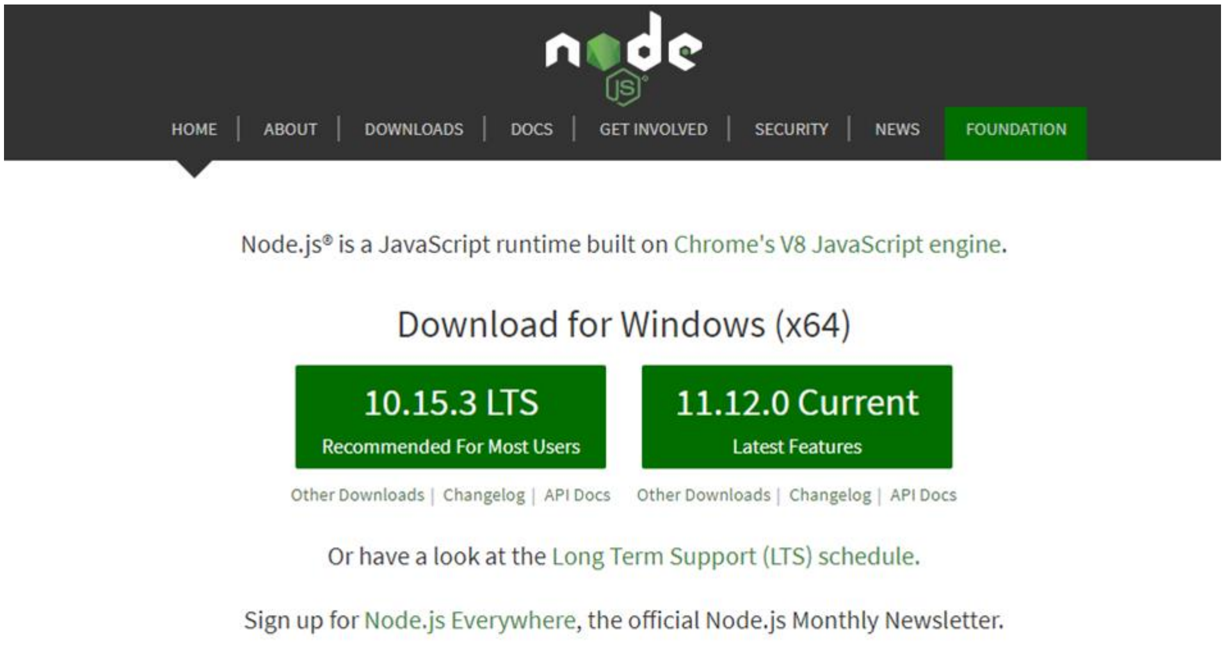


Fig. 7 Depositing Cash sequence



Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

Download for Windows (x64)

10.15.3 LTS
Recommended For Most Users

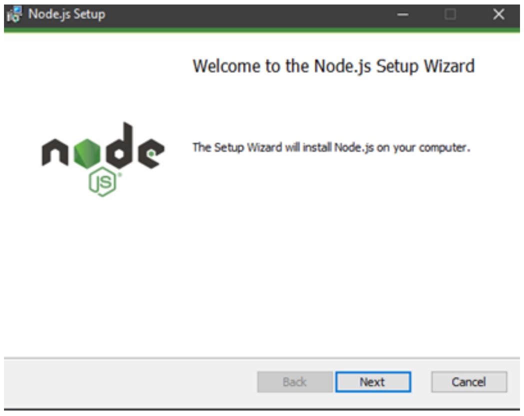
11.12.0 Current
Latest Features

Other Downloads | Changelog | API Docs    Other Downloads | Changelog | API Docs

Or have a look at the Long Term Support (LTS) schedule.

Sign up for Node.js Everywhere, the official Node.js Monthly Newsletter.

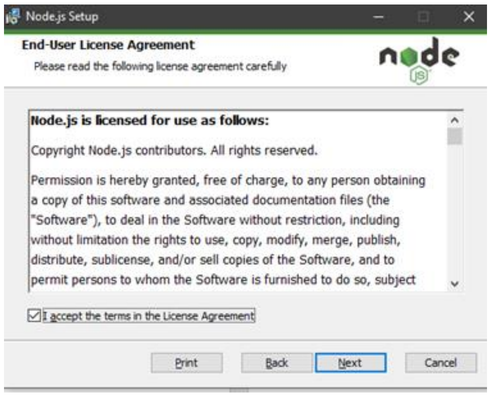
Fig. 8 Downloading Node



Welcome to the Node.js Setup Wizard

The Setup Wizard will install Node.js on your computer.

Fig. 9 Prompts that pop up


Node.js Setup

End-User License Agreement

Please read the following license agreement carefully

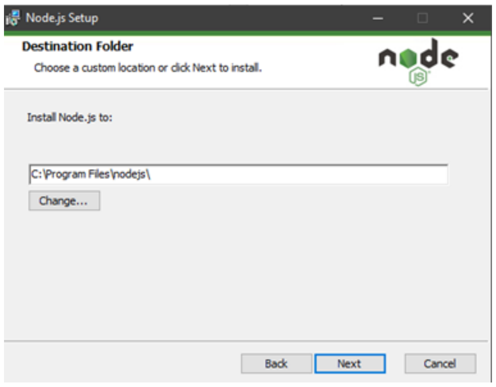
**Node.js is licensed for use as follows:**

Copyright Node.js contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject

☑ I accept the terms in the License Agreement

Fig. 10 Prompts that pop up


Node.js Setup

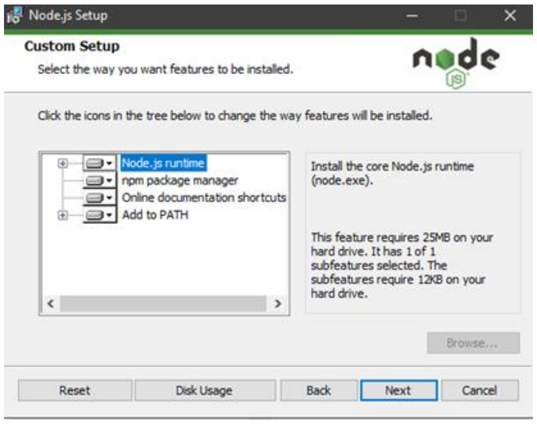
Destination Folder

Choose a custom location or click Next to install.

Install Node.js to:

C:\Program Files\nodejs\

Fig. 11 Prompts that pop up


Node.js Setup

Custom Setup

Select the way you want features to be installed.

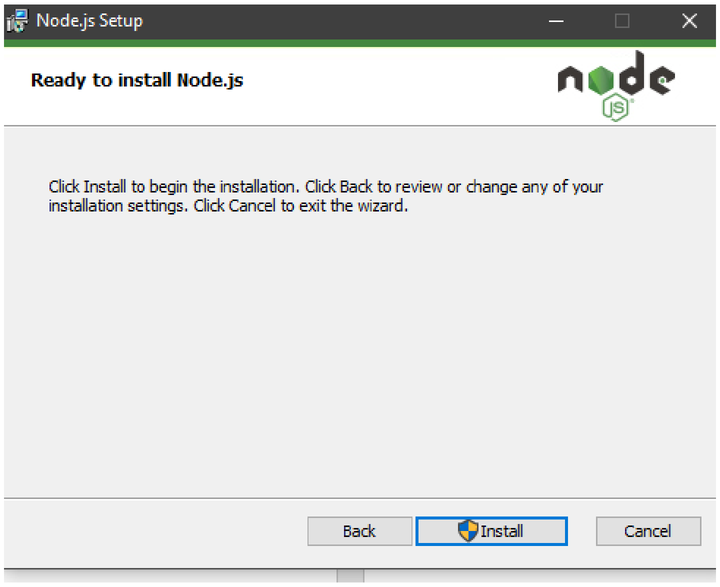
Click the icons in the tree below to change the way features will be installed.

- Node.js runtime
- npm package manager
- Online documentation shortcuts
- Add to PATH

Install the core Node.js runtime (node.exe).

This feature requires 25MB on your hard drive. It has 1 of 1 subfeatures selected. The subfeatures require 12KB on your hard drive.

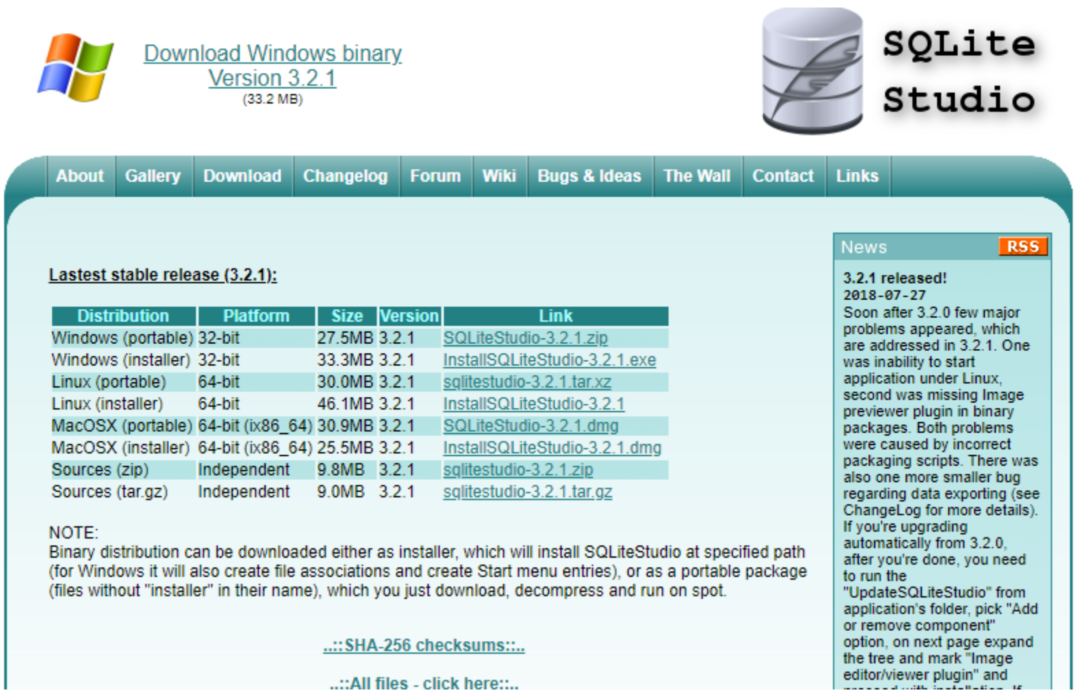
Fig. 12 Prompts that pop up



**Ready to install Node.js**

Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.

Fig. 13 Installing Node.js



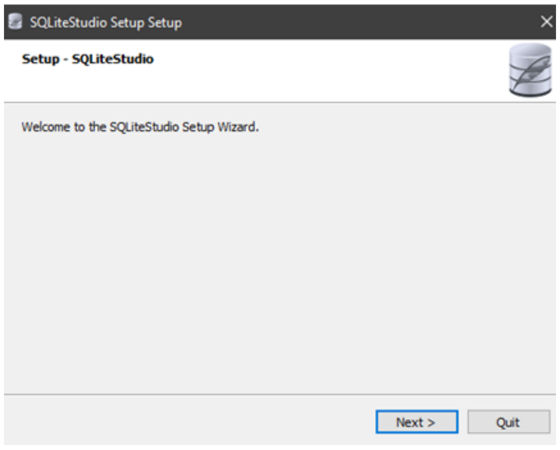
**Lastest stable release (3.2.1):**

| Distribution | Platform | Size | Version | Link |
|---|---|---|---|---|
| Windows (portable) | 32-bit | 27.5MB | 3.2.1 | SQLiteStudio-3.2.1.zip |
| Windows (installer) | 32-bit | 33.3MB | 3.2.1 | InstallSQLiteStudio-3.2.1.exe |
| Linux (portable) | 64-bit | 30.0MB | 3.2.1 | sqlitestudio-3.2.1.tar.xz |
| Linux (installer) | 64-bit | 46.1MB | 3.2.1 | InstallSQLiteStudio-3.2.1 |
| MacOSX (portable) | 64-bit (ix86_64) | 30.9MB | 3.2.1 | SQLiteStudio-3.2.1.dmg |
| MacOSX (installer) | 64-bit (ix86_64) | 25.5MB | 3.2.1 | InstallSQLiteStudio-3.2.1.dmg |
| Sources (zip) | Independent | 9.8MB | 3.2.1 | sqlitestudio-3.2.1.zip |
| Sources (tar.gz) | Independent | 9.0MB | 3.2.1 | sqlitestudio-3.2.1.tar.gz |

NOTE:
Binary distribution can be downloaded either as installer, which will install SQLiteStudio at specified path (for Windows it will also create file associations and create Start menu entries), or as a portable package (files without "installer" in their name), which you just download, decompress and run on spot.

**3.2.1 released!**
2018-07-27
Soon after 3.2.0 few major problems appeared, which are addressed in 3.2.1. One was inability to start application under Linux, second was missing Image previewer plugin in binary packages. Both problems were caused by incorrect packaging scripts. There was also one more smaller bug regarding data exporting (see ChangeLog for more details). If you're upgrading automatically from 3.2.0, after you're done, you need to run the "UpdateSQLiteStudio" from application's folder, pick "Add or remove component" option, on next page expand the tree and mark "Image editor/viewer plugin" and

Fig. 14 Installation

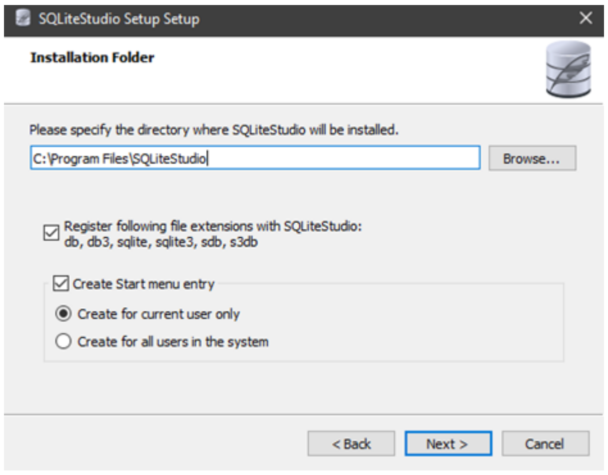


SQLiteStudio Setup Setup

Setup - SQLiteStudio

Welcome to the SQLiteStudio Setup Wizard.

Fig. 15 Following Wizard



SQLiteStudio Setup Setup

Installation Folder

Please specify the directory where SQLiteStudio will be installed.

C:\Program Files\SQLiteStudio

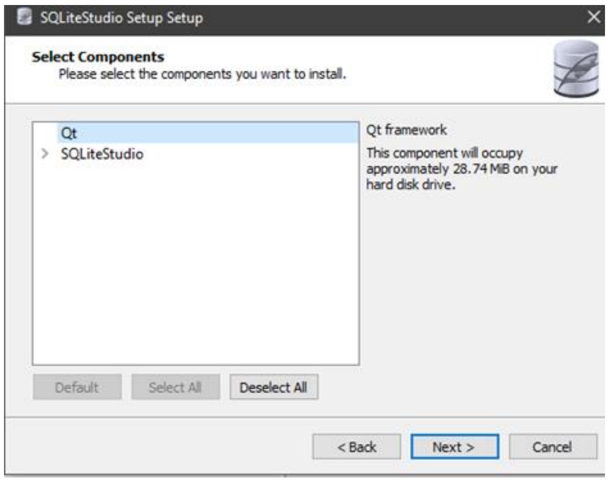
[✓] Register following file extensions with SQLiteStudio: db, db3, sqlite, sqlite3, sdb, s3db

[✓] Create Start menu entry

(●) Create for current user only

( ) Create for all users in the system

Fig. 16 Following Wizard



SQLiteStudio Setup Setup

Select Components

Please select the components you want to install.

Qt

SQLiteStudio

Qt framework

This component will occupy approximately 28.74 MiB on your hard disk drive.

Fig. 17 Following Wizard

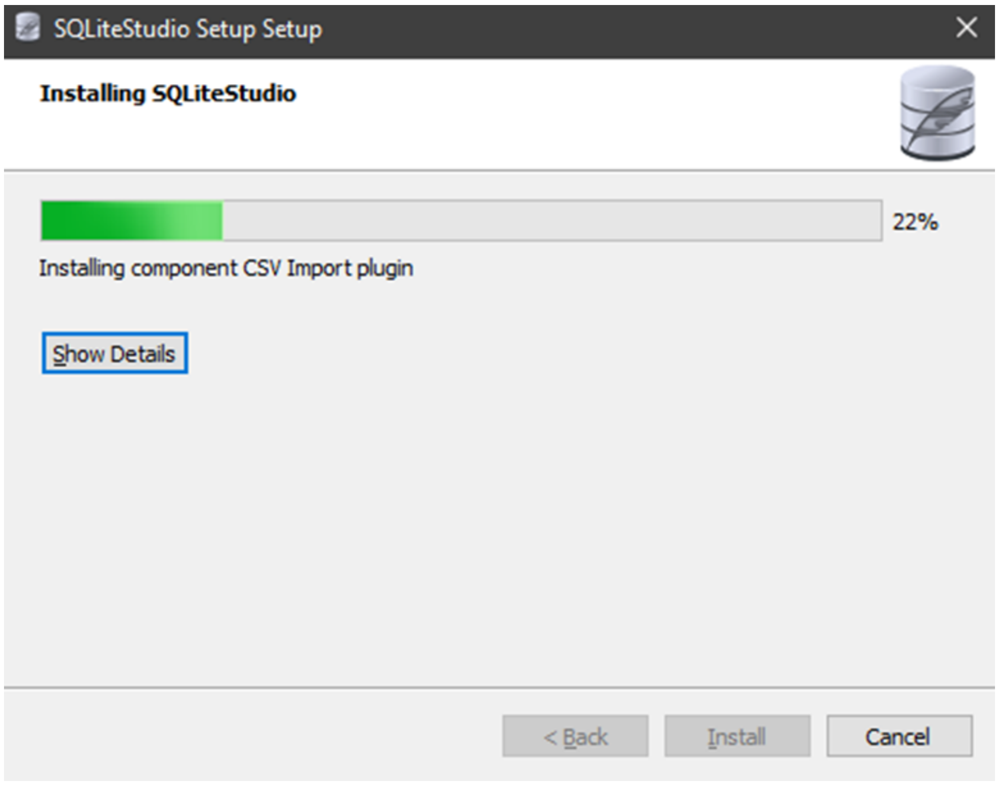


Fig. 18 Installation

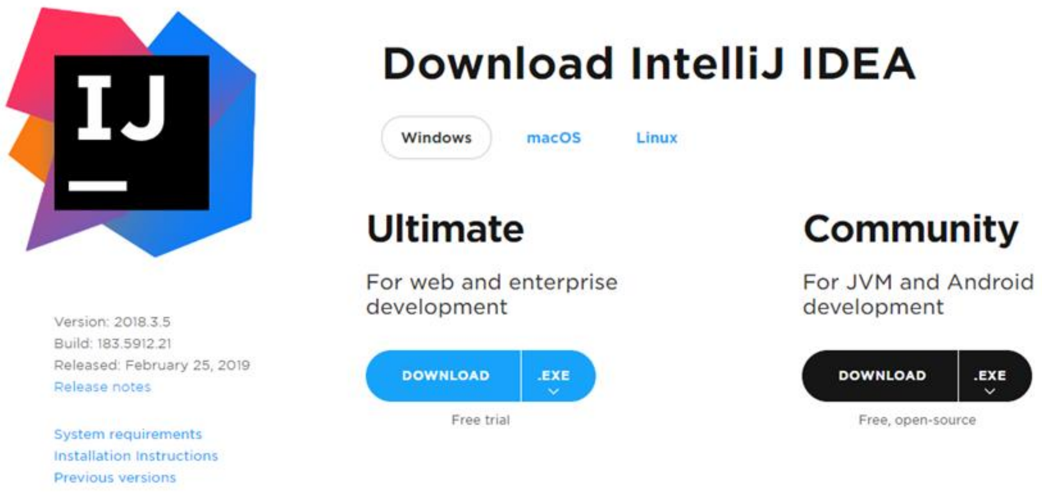
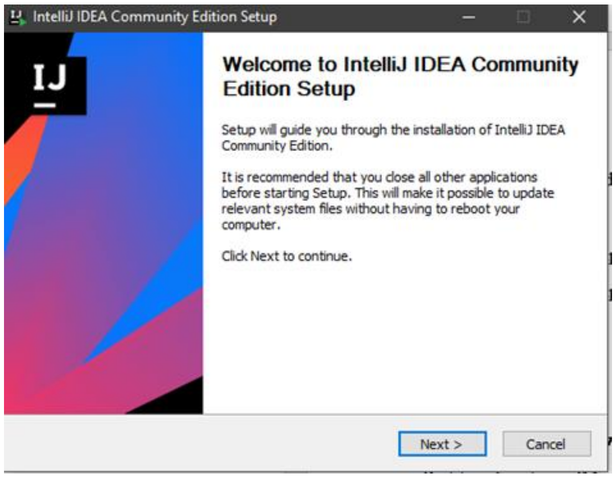


Fig. 19 Installing IntelliJ IDEA


IntelliJ IDEA Community Edition Setup

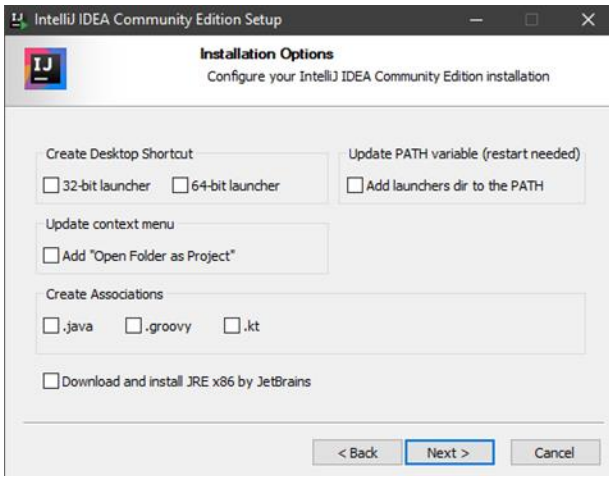
**Welcome to IntelliJ IDEA Community Edition Setup**

Setup will guide you through the installation of IntelliJ IDEA Community Edition.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Fig. 20 Installing IntelliJ


IntelliJ IDEA Community Edition Setup

**Installation Options**

Configure your IntelliJ IDEA Community Edition installation

Create Desktop Shortcut
- [ ] 32-bit launcher
- [ ] 64-bit launcher

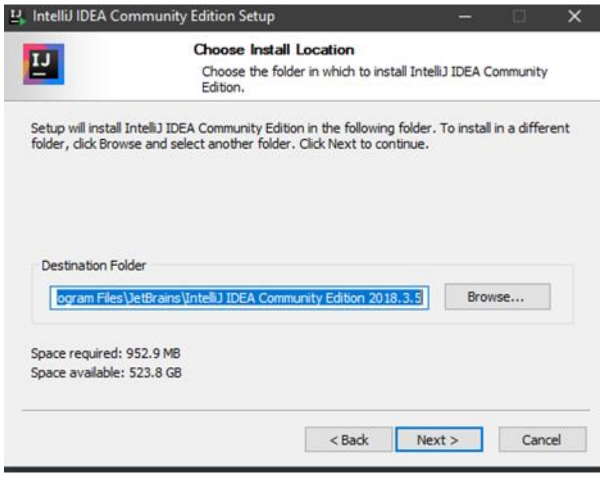
Update PATH variable (restart needed)
- [ ] Add launchers dir to the PATH

Update context menu
- [ ] Add "Open Folder as Project"

Create Associations
- [ ] .java
- [ ] .groovy
- [ ] .kt

- [ ] Download and install JRE x86 by JetBrains

Fig. 21 Installing IntelliJ


IntelliJ IDEA Community Edition Setup

**Choose Install Location**

Choose the folder in which to install IntelliJ IDEA Community Edition.

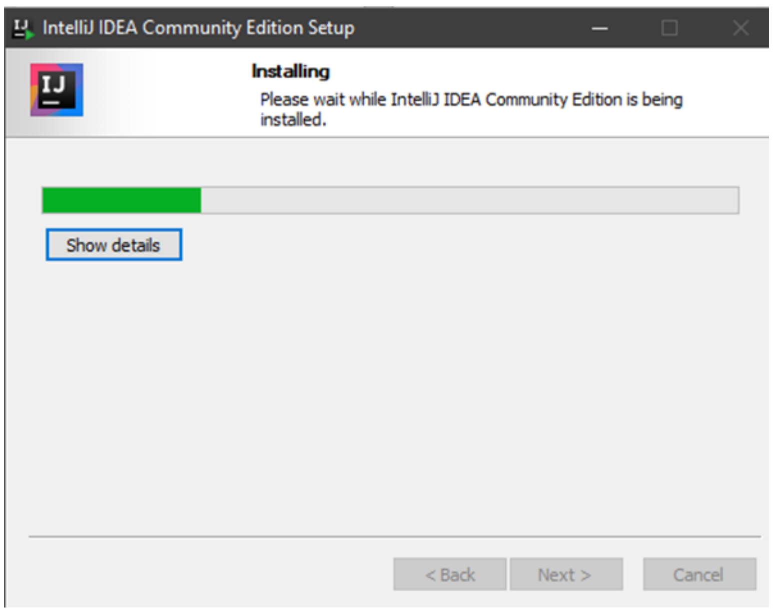
Setup will install IntelliJ IDEA Community Edition in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder

ogram Files\JetBrains\IntelliJ IDEA Community Edition 2018.3.5

Space required: 952.9 MB
Space available: 523.8 GB

Fig. 22 Installing IntelliJ


IntelliJ IDEA Community Edition Setup

**Installing**

Please wait while IntelliJ IDEA Community Edition is being installed.

Fig. 23 Installing IntelliJ