

# UNIVERSITY OF OSLO

## Faculty of Mathematics and Natural Sciences

**Exam in:** INF 4121

**Day of exam:** May 31<sup>th</sup>, 2016

**Exam hours:** 14:30-18:30

**This examination paper consists of 8 pages.**

**Permitted materials:** None

**Answers are permitted in both English and Norwegian**

*Make sure that your copy of this examination paper is complete before answering. Make your own assumptions if you find the description of the exercise insufficient or you want to add something or make changes. Justify your assumptions and changes.*

### **Task 1: Open-ended questions (15 points)**

**Note:**

- For the multiple-choice questions, only one possible answer is correct and can be selected. If you circle 0, 2 or more answers, the question will get zero points.
- For the pair-up questions, draw the arrows corresponding to your choice of pairing. There is only one correct way to do the pairing.

1. According to the ISTQB Glossary, regression testing is required for what purpose?
  - a) To verify the success of corrective actions.
  - b) To motivate better unit testing by the programmers.
  - c) To prevent a task from being incorrectly considered completed.
  - d) To ensure that defects have not been introduced by a modification.

**Answer: D**

2. Pair the following processes with their corresponding activities:

1. Test analysis	A. Group tests into scripts
2. Test plan	B. Write or extract a test summary report for the stakeholders
3. Test implementation	C. Transform the test objectives into test conditions and test cases
4. Test reporting	D. Establish the scope, objectives and risks of testing

**Answer: 1c 2d 3a 4b**

3. Which of the following is a non-functional quality characteristic?

- a) Accuracy
- b) Reliability
- c) Security
- d) Regression

**Answer: B**

4. Which of these is a functional test?

- a) Measuring how long it takes to upload a file to a cloud-based solution.
- b) Checking the effect of high volumes of traffic in a call-center system.
- c) Counting if the number of outputs is as expected.
- d) Checking how easy is to use the software system.

**Answer: C**

5. Acceptance testing is not the responsibility of the development team. It is the responsibility of the customers, but the development team can assist in the process.

- a) True
- b) False

**Answer: A (true)**

6. Static code analysis typically identifies all but one of the following problems. Which is it?

- a) Unreachable code
- b) Memory leaks
- c) Violations of programming standards
- d) Variables that are declared, but never used

**Answer: B**

7. In the \_\_\_\_\_ phase of a review activity, the moderator checks if the defects reported by the reviewers have been indeed fixed by the document author.

**Answer: follow-up**

8. Should pre-conditions and post-conditions be part of a test case?

- a) Yes
- b) No

**Answer: A (yes)**

9. \_\_\_\_\_ is applied as test design technique when the inputs and the outputs of a software under test can be grouped in a way that exhibits similar behavior.

**Answer: equivalence partitioning**

10. Pair the following roles with their typical activities:

Tester	1. Writes automated tests
	2. Gives recommendations to continue or stop the testing, based on the test execution results
	3. Introduces metrics for measuring the test progress
Test leader	4. Acquires and prepares test data
	5. Writes test summary reports for management
	6. Evaluates the results of the execution of tests: pass or fail

**Answer:**

**Tester 1, 4, 6**

**Test leader: 2, 3, 5**

11. Which of the following metrics would be most useful to monitor during test execution?

- a) Number of testers versus number of developers in the team.
- b) Number of defects found and fixed.
- c) Number of test environments that should have been set up, but were abandoned.
- d) Percentage of requirements for which a test has been written.

**Answer: B**

12. Is it allowed to use low-fidelity prototyping when designing a user-centric software system?

- a) Yes
- b) No

**Answer: A (yes)**

13. What does it mean that a user interface is operable?

- I. All functionality can be operated by the same user.
  - II. All functionality is available from a keyboard.
  - III. All user interface is navigable with a keyboard.
- a) I, II
  - b) I, III

- c) II, III
- d) I, II, III

**Answer: c) II, III**

14. \_\_\_\_\_ is a design issue of software for mobile devices. It occurs when the user touches by mistake some elements (links, buttons) or makes a gesture that unexpectedly initiates a feature.

**Answer: accidental activation**

15. Works in the \_\_\_\_\_ can be reproduced or used by anyone.

**Answer: public domain**

### **Task 2: Short questions (10 points)**

1. What are the five fundamental test activities? Briefly describe their respective tasks from planning to closure.
  - **Plan & control**
  - **Analysis and design**
  - **Implementation & execution**
  - **Reporting & exit criteria**
  - **Test closure activities**
2. Explain briefly the following terms:
  - regression testing: **unchanged** areas of software perform as expected after a bugfix.
  - confirmation testing: test again a **changed** area of software to confirm that the bug has been fixed.
3. Identify and describe three types of non-functional software characteristics.
  - **reliability**, which is defined further into the sub-characteristics maturity (robustness), fault-tolerance, recoverability and compliance;
  - **usability**, which is divided into the sub-characteristics understandability, learnability, operability, attractiveness and compliance;
  - **efficiency**, which is divided into time behavior (performance), resource utilization and compliance;
  - **maintainability**, which consists of five sub-characteristics: analyzability, changeability, stability, testability and compliance;
  - **portability**, which also consists of five sub-characteristics: adaptability, installability, co-existence, replaceability and compliance.
4. How does testing depend on the development life-cycle for the software under test? (sequential and iterative-incremental)  
**The V-model** provides guidance → testing needs to begin as early as possible in the life cycle  
Testing should be carried out in parallel with development activities, and testers need to work with developers and business analysts so they can perform these activities and tasks and produce a set of test deliverables.

V-model uses four test levels:

- component testing
- integration testing
- system testing
- acceptance testing

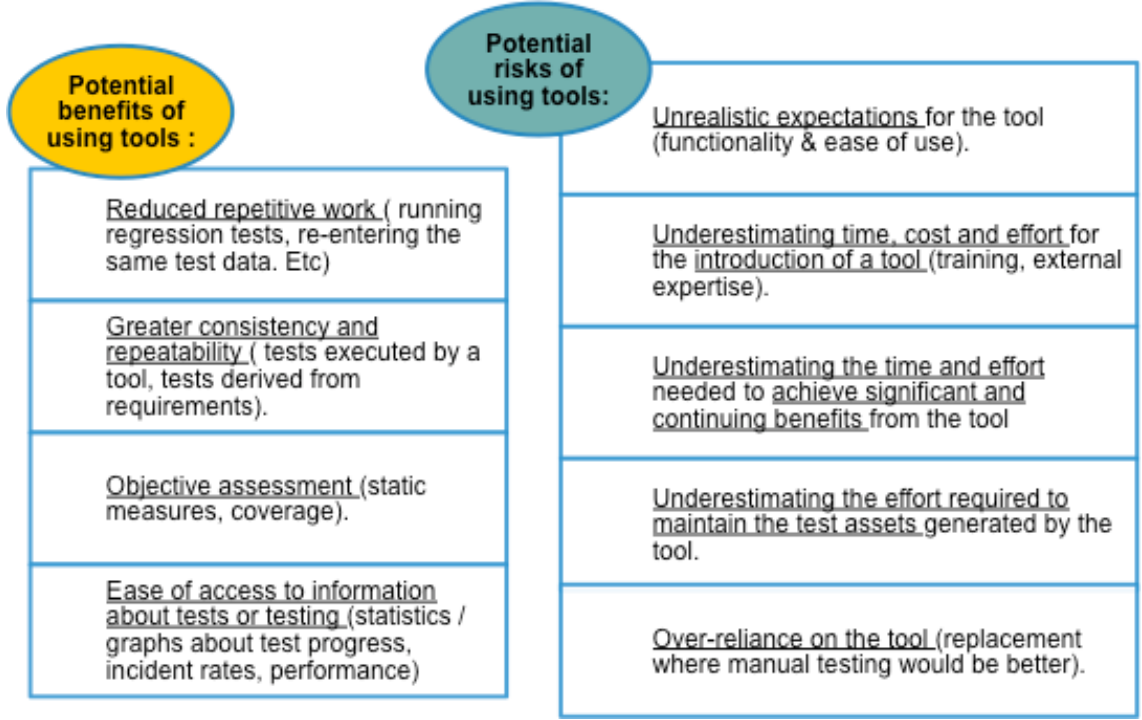
**Iterative-incremental:**

- **Regression testing is increasingly important on all iterations after the first one.** This means that **more testing will be required** at each subsequent delivery phase which must be allowed for in the project plans

5. Define and explain the purpose of entry criteria and exit criteria in software testing.

**Entry criteria** is the minimum set of conditions that should be met in order to start the testing work  
**Exit criteria** are the set of conditions that should be met in order to close a particular testing phase.

6. Summarize the potential benefits and potential risks a company may face when using test tools.



7. What is a decision table? Provide an example.

Decision tables are a good way to:

- capture system requirements that contain *logical conditions*,
- and to document internal system design.

The input conditions and actions are most often stated in such a way that they can either be true or false (Boolean).

The strength of decision table testing is that it creates combinations of conditions that might not otherwise have been exercised during testing. It may be applied to all situations when the action of the software depends on several logical decisions.

	Rule 1	Rule 2	Rule 3	Rule 4
Conditions				

Oslo resident?	False	True	True	True
Over 18 years?	Don't care	False	True	True
Happy hour?	Don't care	Don't care	False	True
<b>Actions</b>				
Can buy wine?	False	False	True	True
Offer 10% discount?	False	False	False	True

8. Explain the role of the personas in the study of accessibility. Provide an example of such a persona.

The personas can help address big challenges in approaching the usability issues:

- give a realistic view of the people we design for
- help taking different users into account (will tell a story we can relate to)
- help organizing increasing amounts of data; will document our assumptions
- build consensus around a clear, consistent view on accessibility needs to be solved

Example:

- autism
- cerebral palsy
- blindness
- fatigue
- deaf-mute
- age-related macular degeneration
- non-english speaker

9. Provide two examples of issues that we may encounter with software running on mobile devices.

- **Low discoverability:**  
The interactive part is mostly hidden within the design.
- **Low memorability:**  
Gestures are not consistent across applications and are hard to remember. They can be confusing for both the users and the device (as are all recognition-based user interfaces).
- **Accidental activation:**  
This occurs when users touch by mistake some elements or make a gesture which unexpectedly initiates a feature.
- **Download times dominates the user experience.**  
Most pages take far too long to load.  
Even the high-end phones deliver much slower browsing than desktop computers.  
As result: users are reluctant to request additional pages and they give up easily.
- **Scrolling causes major usability problems, especially in the sites not optimized for mobiles**  
The main problem is that the pages scroll too much.  
As result: people lose track of where they were and what info is in the page. Also people scroll right past something, without noticing it.
- **Zooming**  
Zooming into a page allows users to read more easily.
- **Bloated pages**  
They make users lose context and feel lost.  
On computers these pages normally don't feel bloated, but rendered on a mobile they bloat.

- **Mobile reading is harder**

Users can see less at any given time. They have to rely on their short-term memory when they are trying to understand anything that is not fully explained within the viewable reading space.

Users have to move around the page more, scrolling to refer to other parts of content, instead of simply glancing at the text.

Mobile users are in a hurry and get visibly angry at sites that waste their time.

Bullet points make information more scannable on a small screen

10. Describe the notions of:

- public domain
- fair use.

The term “**public domain**” refers to creative materials that are not protected by intellectual property laws such as copyright, trademark, or patent laws.

**Fair use** is any copying of copyrighted material done for a limited and “transformative” purpose, such as to comment upon, criticize, or parody a copyrighted work. Such **uses** can be done without permission from the copyright owner.

### ***Task 3: Problems to solve (15 points)***

#### **Problem 1**

A vending machine dispenses either hot or cold drinks. If you choose a hot drink (e.g. tea or coffee), it asks if you want milk (and adds milk if required), then it asks if you want sugar (and adds sugar if required), then your drink is dispensed.

Draw a control flow diagram for this example.

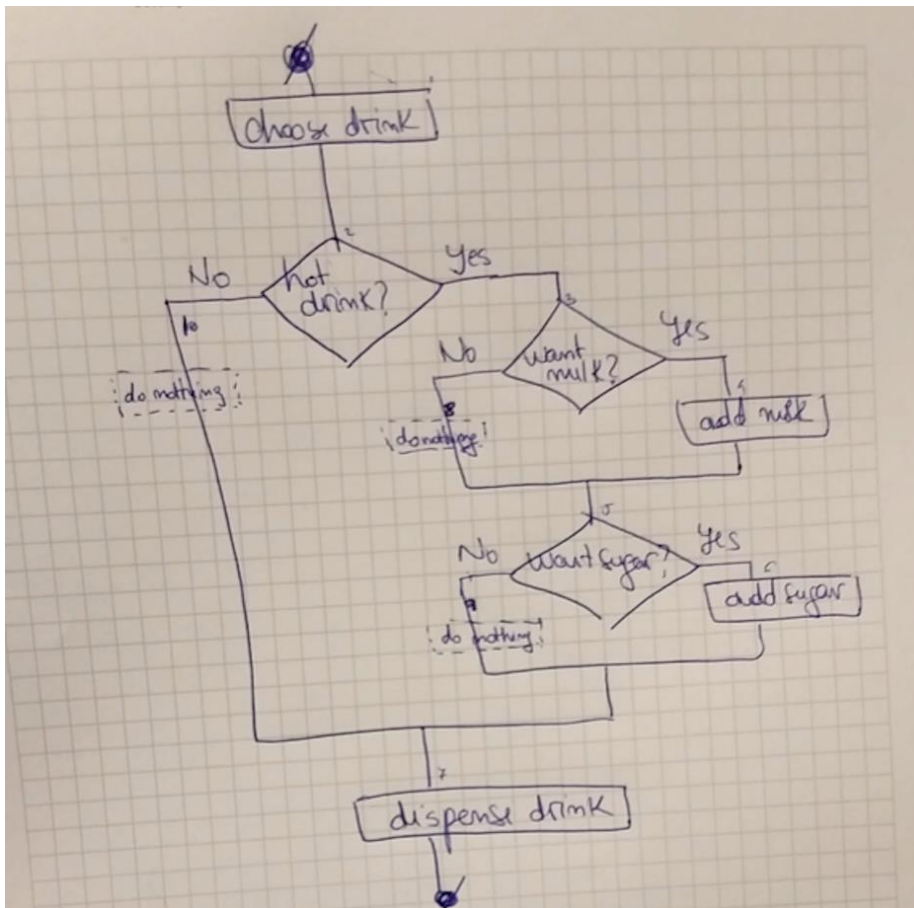
Given the following tests, what is the statement coverage achieved for the following tests:

- Test 1: Cold drink
- Test 2: Black coffee (no milk, no sugar)

What additional tests would be needed to achieve 100% statement coverage?

Statement coverage is : 5/7

Additional test: hot drink with sugar and milk



## Problem 2

You have a savings account (Spar) in a bank that gives you a different interest rate, depending on the how much money you have in that account.

- if you have more than 0 NOK and up to 1.000,00 NOK (included), you will get 2% interest rate per year
- if you have more than 1.000,00 NOK and up to 10.000,00 NOK (included), you will get 3% interest rate per year
- if you have more than 10.000,00 NOK, you will get 4% interest rate per year

Note: it's not allowed to have a negative amount of money in the account.

Assume that after one year you decide to get rid of your savings account and open another super-savings account (Super-Spar) in the same bank. This second account works a bit different:

- If you have less than 5.000,00 NOK in the account, it will give you 5% interest rate per year.
- If you have more than 5.000,00 NOK in the account, it will give you 10% interest rate per year.
- Note: it's not allowed to have a negative amount of money in the account.

Which test design technique should you chose to check if both bank accounts give you correct interest rates?

Which is the minimum number of tests you need in order to completely check if both accounts give you correct interest rates? Please include in the counting the invalid cases as well.

Provide for each test listed above an example with a relevant input value.

### Test design technique:

BVA (boundary value analysis) EP (equivalence partitioning) is OK too.



**Number of tests: 5 in total** (3 for spar, 2 for super spar)

**Example of tests:**

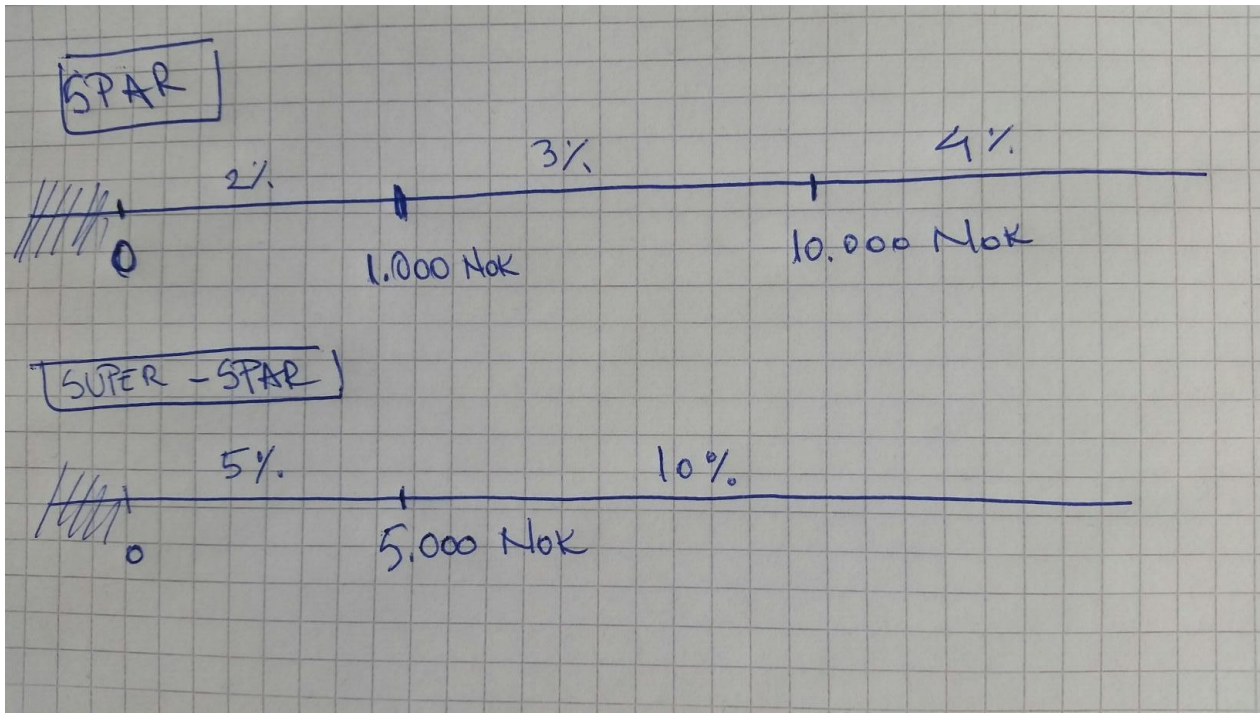
Test 1: spar: 500 NOK

Test 2: spar: 8000 NOK

Test 3: spar: 12000 NOK

Test 4: super spar: 4000 NOK

Test 5: super spar: 6000 NOK



### **Problem 3**

Given the following extract of a requirement document for an e-commerce website:

Req.08: User must be able to add items to his shopping cart and continues shopping

**Pre-condition:**

The user is logged-in the system.

The user already has a few items in the cart.

**Main branch:**

1. The user must be able to browse the shopping list, select items and click "Add to cart".
2. The system must update the user's shopping cart accordingly (items, quantity, price)
3. The warehouse must decrease the inventory is the items added to the cart.
4. The user must be logged out of the website.

**Post-condition:**

The user is logged out. The cart is empty.

Identify the issue in the use case scenario above and make an incident report (a.k.a. anomaly report) for it. Include in the incident report the information required cf. IEEE 829, presented in Lecture 05. If there are fields irrelevant to the issue, then fill them with “non applicable”.

Details of the incident report may include (cf. IEEE 829):

**Date:** 31.05.2016

**Project:** E-commerce website

**Programmer:** not assigned

**Tester:** candidate 4012

**Program/Module:** E-commerce website

**Build/Revision/Release:** not applicable

**Software Environment:** not applicable

**Hardware Environment:** not applicable

**Status of the incident** new (newly reported)

**Number of Occurrences:** 1

**Severity:** major / minor

**Impact** minor

**Priority** n.a.

**Detailed Description:** WHEN A USER SELECTS AN ITEM AND ADDS IT TO THE SHOPPING CART, HE SHOULD NOT BE LOGGED-OUT.

**Expected result:** The user should be able to continue shopping or go to check-out, without being logged out

**Actual result:** The user is logged out after adding an item to the shopping cart.

**Change history:** not applicable

**References:** not relevant

**Assigned To:** not defined yet

**Incident Resolution:** not defined yet

#### **Task 4: Essay-type questions (10 points)**

### **Test automation:**

Provide a simple explanation for the process of test automation: a brief definition, the context in which it is used, discuss why is it important.

List the different interfaces for automated testing. Explain what can be tested through each.

Explain and compare the advantages and limitations of test automation.

List the main approaches for creating automated test cases

List the major success factors in test automation.

### **Definition and use:**

In software **testing**, **test automation** is the use of special software (separate from the software being tested) to control the execution of **tests** and the comparison of actual outcomes with predicted outcomes.

Test automation can automate some repetitive but necessary tasks in a formalized testing process already in place, or perform additional testing that would be difficult to do manually.

### **Interfaces: GUI and API tests:**

**Graphical user interface testing.** A testing framework generates user interface events such as keystrokes and mouse clicks, and observes the changes that result in the user interface, to validate that the observable behavior of the program is correct.

**API driven testing.** A testing framework that uses a programming interface to the application to validate the behaviour under test. Typically API driven testing bypasses application user interface altogether. It can also be testing public (usually) interfaces to classes, modules or libraries are tested with a variety of input arguments to validate that the results that are returned are correct.

### **Advantages:**

- More tests are run
- Tests that cannot be done manually are enabled (real-time, remote, parallel tests)
- Tests can be more complex
- Tests run faster
- Tests are less subject to operator error
- More effective and efficient use of testers
- Improved system reliability
- Improved quality of test

### **Limitations:**

- One cannot automate all manual tests
- The automation can only check machine-interpretable results
- The automation can only check results that can be predicted or verified

### Approaches:

- capture and replay
- data-driven
- keyword-driven

### Success factors:

- The system under test must be designed for testing.  
Invariably, the SUT needs some changes to support automated testing or to allow new types of tests to be executed against it. This is an essential part of success with test automation.
  - UI testing: the SUT should decouple as much as possible the GUI interaction data from the appearance of the graphical interface.
  - API testing: more classes or modules may need to be exposed as public so that they can be tested.
- Good Test Automation Architecture (TA-A): The architecture of the test automation is just as the architecture of a software work product, and should be treated as such.
- Target the testable parts of the SUT first. Generally, a key factor in the success of the test automation lies in the ease of implementing automated test scenarios.
- Creation and maintenance of a test automation framework (TA-F): The framework should be easy to use, well documented, and maintainable.

## Manual testing of an ATM

An automated teller machine (ATM) is an electronic banking outlet, which allows customers to complete basic transactions without the aid of a bank employee.

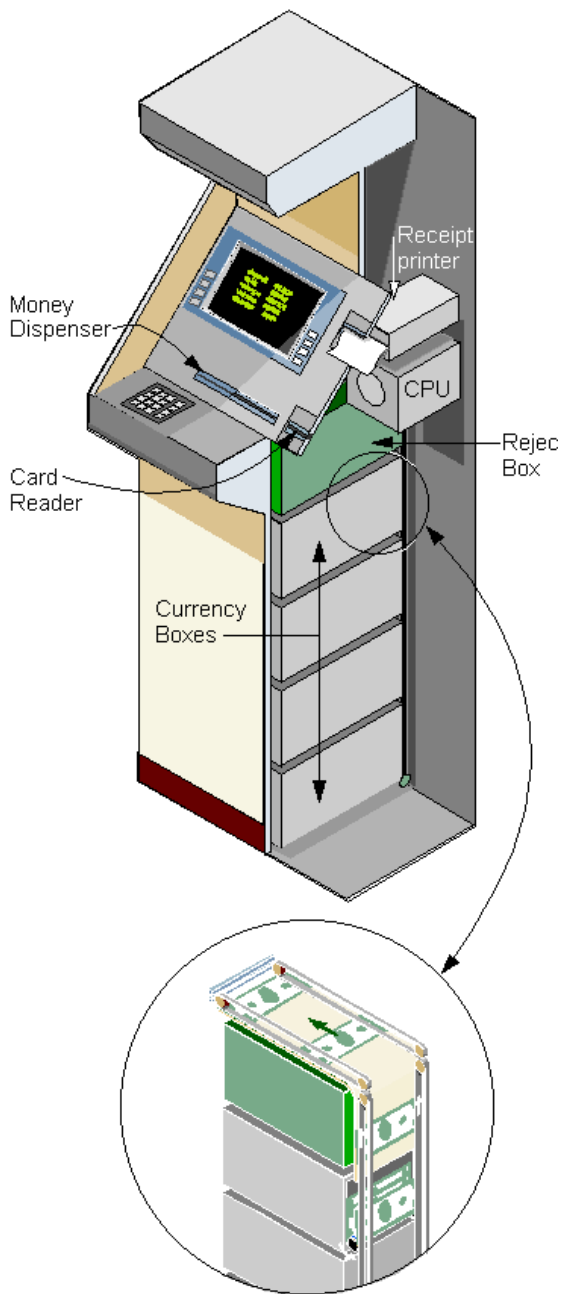
Banks operate ATMs and customers can access them, even if they are or not customers of the bank owning the ATM. All the customers need is a valid bank card. The ATMs need to communicate with the bank's servers, in order to respond to customer's enquiries and actions.

There are two primary types of automated teller machines, or ATMs. The basic units allow the customers to only withdraw cash and receive a report of the account's balance. The more complex machines will accept deposits, facilitate credit card payments and report account information.

For the purpose of this exercise, we will only consider the basic type of ATMs.

Requirements:

- Imagine a version of such an ATM and write a brief description for it.
- Create a list with the manual tests that you would run on the ATM described at the previous point. Specify the actors that you took into consideration for your testing.
- Prioritize the list of tests created above, in function of their importance.
- Specify what you have left out of your list of tests and why.



## Description of the ATM

### Test Scenarios of ATM Machine

1. Verify that user is presented with option to choose language for further operations
2. Verify that user asked to enter pin number before displaying any card/bank account detail
3. Verify that there are limited number of attempts upto which user is allowed to enter pin code
4. Verify that if total number of incorrect pin attempts gets surpassed then user is not allowed to continue further- operations like blocking of card etc gets initiated
5. Verify that user is presented with different account type options like- saving, current etc
6. Verify that user is allowed to get account details like available balance

7. Verify that user same amount of money gets dispatched as entered by user for cash withdrawal
8. Verify that user is only allowed to enter amount in multiples of denominations as per the specifications
9. Verify that user is prompted to enter the amount again in case amount entered is not as per the specification and proper message should be displayed for the same
10. Verify that user cannot fetch more amount than the total available balance
11. Verify that user is provided the option to print the transaction/enquiry
12. Verify that user user's session timeout is maintained and is as per the specifications
13. Verify that user is not allowed to exceed one transaction limit amount
14. Verify that user is not allowed to exceed one day transaction limit amount
15. Verify that user is allowed to do only one transaction per pin request
16. Verify that user is not allowed to proceed with expired ATM card
17. Verify that in case ATM machine runs out of money, proper message is displayed to user
18. Verify that in case sudden electricity loss in between the operation, the transaction is marked as null and amount is not withdrawn from user's account

**Most important:**

**7, 8, 10, 17, 2, 3, 4, 13, 16**

**Left out:**

- All non-functional test: robustness, suitability, accuracy, effectiveness, etc.
- Encryption of communication
- Verify that font of the text displayed in ATM screen is as per the specifications
- Verify that touch of the ATM screen is smooth and operational