Test design techniques overview.
White Box techniques
Assignment

Stefania Pruteanu

Level 1: Make a comparison of static and dynamic testing techniques. Give the advantages and possible limitations when using each of them.

	Static testing technique	Dynamic testing technique
Description	does not require running a program or application and allows to find the most obvious errors at the early stages of product creation.	is a type of testing that involves running the program code. That is, the behavior of the program is analyzed during its operation.
Advantage 1	Early testing before implementation	Early testing before product's launch
Advantage 2	Identify errors with requirements and code	Detecting system defects and failures
Advantage 3 (etc.)	Closer team collaboration	Broad results, various functional & non functional tests
Limitation 1	Might postpone development's time	Can postpone launch time
Limitation 2	Can't predict UX, system's behaviors	May need costly resources (environment , hardware , software)
Limitation 3 (etc.)	Leads undetected errors to running defects	Relies on static testing's outcomes
Conclusion	Best for writing code, requirements reviews, test plans	Best for UAT, functional & non-functional test runs

2. We have Elevator Control System, which has the following characteristics:

1) Floor numbers: 1 to 10 (inclusive).
2) If the requested floor number is outside this range, the system displays an error message.
3) If the requested floor number is within the valid range, the system moves the elevator to that floor.

Create the test cases based on EP in TestRail under your project

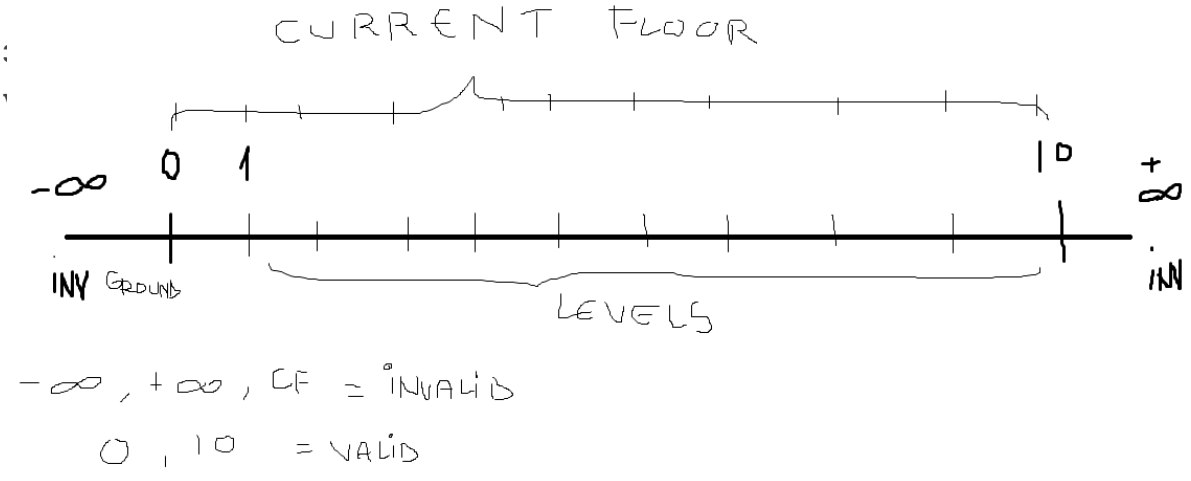
I made a pseudocode to visualize this:

```
INT A,B, FLOOR,  CURRENTFLOOR
READ A,B
DIGITS = A & B
CURRENTFLOOR = DIGITS;    (but here's a separate function of how digits will be added together..)

IF
        CURRENTFLOOR > 10
        {
        WRITE "Floor unavailable" }

ELSE IF CURRENTFLOOR = (0,10)
                {
                WRITE " Going to CURRENTFLOOR "; }
        ELSE
        { WRITE " You are on CURRENTFLOOR "; }
```

EP:



Explaining my messy drawing:

STATES (floor levels) :

VALID

1. SINGLE DIGITS -

0- ground level

1-9 - higher levels

2. DOUBLE DIGITS -

0X - X is any number, *in some cases* people do enter the numbers as 01,02.. on an elevator so they should be validated

10 - the exceptional case where a double digit could be valid for a 10-level elevator

INVALID

3. DOUBLE DIGITS -

> 10 - those should not be accepted

DECISIONS (going to floor levels):

VALID

1. Going to valid floor level

INVALID

***2.* Not going to invalid floor level**

3. Not going to the same valid floor level

So to simplify this :

The system will not take any numbers with more than 2 digits:

- Users can travel to any different level within valid range (0-10)

- Users cannot travel to the same level within valid range (0-10)

- Users cannot travel to any invalid range (>10)

^ 3 test cases, assuming the elevator makes the digits calculations the right way

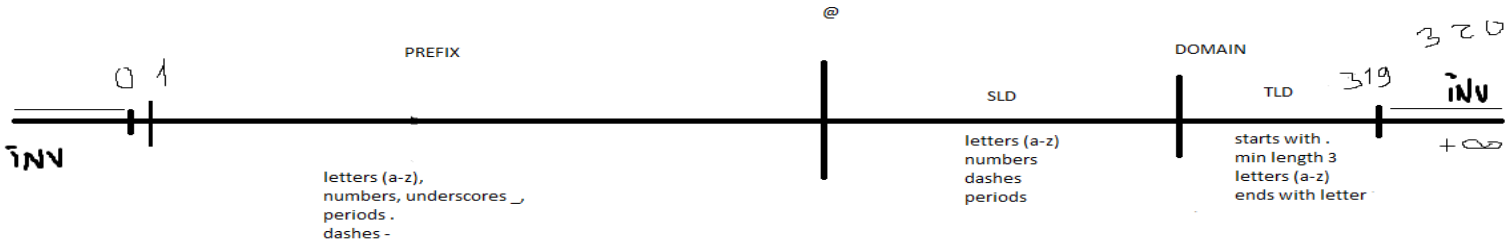
100% coverage for decisions?

3. Apply Equivalence Partitioning to validate email addresses in a web application. The email validation criteria are as follows:

1) **Valid email format**: The email address should follow a standard format with the "@" symbol and domain name (e.g., example@example.com).

2) **Maximum Length**: The email address should not exceed 320 characters.

3) **Domain Validation**: The domain name should be valid and should not contain special characters.



	VALID	INVALID
email format / FIELDS	**HAS ALL REQUIRED FIELDS**	**MISSING REQUIRED FIELDS**
PREFIX / username	Letters, numbers, underscores, periods, dashes, starts & ends with letter	Consecutive underscores, periods, dashes (space), special characters
@	@	Duplicates (@..@..)
SLD / subdomain / domain	Letters, numbers, dashes, periods, starts & ends with letter	Consecutive periods, dashes (space), special characters
TLD	Letters Min. length 3, starts with . ends with letter	Consecutive periods (..) special characters, numbers, dashes (space)
TOTAL LENGTH	**< 320**	**0**

For 5 statements (length, prefix , @ , SLD & TLD), and 10 decisions * 2 tests(valid / invalid) for full coverage?