2. User requirements

2.1 User roles

<This section lists the potential users of the product. Get to know the stakeholders for the product and discuss the requirements with them. For each user category list the following information:

- *Name/category: name of user group*
- *User role: responsibilities of the user*
- *Subject-matter experience: users' knowledge of the domain >*

2.2 User stories

<This section describes the main, high-level requirements stated by the users.
User requirements may be a set of statements or use case scenarios presented by the client in layman's terms of which the client can easily elaborate and are usually free of technical jargon.
Consider writing user requirements in the form of user stories. Add an explanation or (reference to an) illustration to describe the requirement. Add acceptance criteria to describe the boundaries to the story and the minimum that is needed to make it acceptable.
*The priority of the requirements is rated according the MoSCoW (**M**ust, **S**hould, **C**ould, **W**on't) rating.>*

ID	Requirement	Description / explanation	Priority
UR.1	The user should be able to start and play a game of 4 in a row against the robotized opponent without operator intervention.		Must
UR.2	The robot should detect a cheating player and respond by resetting the game.		Must
UR.3	The user shall be notified when the game ends.	The system must communicate somehow who won (sounds, visual, giving a coin).	Should
UR.4	The system is able empty the playfield, separate the coins by colour and prepare itself for the next game.		Must
UR.5	After a game, The coins must move to the sorting base, by emptying the game board column by column.	In order to avoid obstruction during clearing the board game and make the coin checking principle easier.	Must
UR.6	From the sorting base, the yellow and red coins shall be sorted and returned to their belonging base.		Must
UR.7	A flipper will shoot the human (yellow) coins back to their base.		Must
UR.8	The robot head shall be controlled to the desired X and Z position within 1.5mm accuracy		Must
UR.9	The insertion of a game coin in an arbitrary column shall be detected by dedicated sensors.		Must
UR.10	Cheating by inserting double coins should also be handled		Must
UR.11	The robot head should suck up tokens by actuating the sucker.	Research needs to be done on the sucking power wrt the coins.	Must
UR.12	The robot head sucker can release the coin at a given position to insert the coin into game.		Must
UR.13	Requirements from Pascal:		
UR.14	A Board Support Package (BSP) must be made of the operating system with which the necessary hardware components of the robot can be controlled.		Must

UR.15	The components of the BSP that are necessary for the functioning of the robot must be tested independently of each other within the project.		Must
UR.16	The algorithm running on the Raspberry Pi must be integrated on the new STM32H7 dual-core.		Could
UR.17	A physical demo must demonstrate that the modular structure of the software architecture is applicable and functioning.		Should

Unit Test Plan

- AI's
- Endswitches
- Homeswitches
- Encoder readout
- Motor control
- Timing interrupt
- PID calculations
- Movements
- Vacuum components
- Coin detector
  - ➔ Multiple coins after another
  - ➔ Multiple coins in different columns
- Servo control
- Flipper control
- RGB sensor
- Emergency stop
- Difficulty setting
- Reset button
- Subfunctions?

Integration Test Plan

- Initialization sequence
- Home procedure
- Normal play sequence (put coin inside column)
- Game end sorting sequence
- Emergency stop and recovery procedure

System Test Plan (use cases?)

- Starting the system
- Playing of multiple games
  - → winning
  - → losing
  - → different difficulties
- Shutdown