

# Requirements 2

(Part of *Change2.pdf*)

---

Cohort 1, Team 4:

Robert Watts  
Adam Wiegand  
Josh Hall  
Travis Gaffney  
Xiaoyu Zou  
Bogdan Lescinschi

# Requirements

When determining the requirements for this software engineering project, we determined 3 key areas to obtain the information required to express the requirements for the project, which involved the product brief, the stakeholders, and our own team decisions.

Our first step was to analyse and quantify the requirements from the product brief provided to us. This involved creating an initial list of all the requirements from the document and discussing the implications of these with the team. As a lot of these requirements were critical to the success of the project and their implementation was required, many have been classified as functional requirements of the system.

Once the requirements from the product brief were categorised and discussed, we then moved onto the second stage, which involved interviewing a stakeholder in order to obtain additional requirements, constraints and expectations of the system. This process was performed by arranging a meeting with the customer, and asking them a set of predetermined questions. These questions were discussed between members of our development team beforehand and were specifically written for achieve a few key objectives:

- Elaborate on requirements from the product brief in order to achieve a greater understanding of the specifics regarding them and any potential constraints.
- Obtain new functional requirements for the project. (additional critical functionality)
- Gather some expectations of the user for the project.
- Understand if and where there was flexibility for deciding our own requirements.

These questions covered a lot of topics regarding the project including the style of the game, a target audience, mechanics and other aspects that previously had not been covered in the product brief. While the interview was being conducted with the customer of the product, members of our team noted down key information / transcripts of the answers provided, in order to ensure we had the correct information provided by the stakeholder. This data was again aggregated, and discussions were made with our team around the answers provided.

Once we had gathered all the required data from the product brief alongside additional prompts from the stakeholders, another discussion was performed in order to determine the final initial requirements of the project, mostly prompted from the answers provided by the customer. These requirements were determined from specifics of the system that the stakeholder expressed to us were our decision to make, and were mostly non-functional requirements, such as how certain features of the game would be implemented.

Taking all of this data into consideration, we have decided to format the provided requirements into 4 tables - consisting of:

- Constraints provided that will have an impact on the development of the system
- The user's requirements and expectations from the system
- The functional system requirements
- The non-functional system requirements

Additionally, this statement of requirements will include criteria for quantifying the success of the requirement.

Constraints	
ID	Description
CON_PLATFORM	Design for a desktop experience, taking into account aspect ratio
CON_PROG_LANG	Must be developed with the Java programming language
CON_REAL_TIME	Game must run in real time (not turn based)
CON_TOP_VIEW	Game must be from a top down view

User Requirements		
ID	Description	
UR_NAME	Game must be named "Auber"	
UR_UX	The user interface should be easy and intuitive to navigate	
UR_MAP	The game displays a map to the user which can be navigated and interacted with	
UR_PLAYER	The player can navigate the map, heal and arrest hostiles	
UR_HOSTILES	Hostiles act like other characters on the ship, but attempt to sabotage key systems and have special abilities	
UR_LOGIC	Game is played in real time, with no turns. Player wins if all hostiles are arrested and loses if more than 15 key systems are destroyed.	
UR_SYSTEMS	There should be 15 systems around the map that hostiles can interact with.	

Functional System Requirements		
ID	Description	User Requirements
FR_MENU	The system should provide a menu to allow the player to start and configure the game	UR_UX

FR_MAP	The system should have a fixed playable map area consisting of at least distinct 4 types of rooms. The map will be made of tiles on which the player and the AI can traverse, from tile to tile	UR_MAP
FR_TELEPORTER	The map should contain teleport pads in areas which will the player can use to teleport around the map	UR_MAP
FR_KEY_SYSTEMS	The map should contain at least 15 key systems which can be sabotaged by infiltrators, and defended by the player. These cannot be repaired.	UR_MAP, UR_SYSTEMS
FR_ALIENS	There must be at least 2 types of alien characters on the map	UR_MAP
FR_HOSTILES	There are eight hostiles on the map	UR_HOSTILES
FR_SABOTAGE	The hostiles can sabotage key systems on the map	UR_HOSTILES
FR_SPECIAL_ABILITIES	The hostiles must have at least 3 special abilities	UR_HOSTILES
FR_PRISON	Holds hostiles once they have been arrested. Arrested hostiles cannot escape.	UR_MAP
FR_ATTACK_NOTIFICATION	The player is notified of sabotage attempts	UR_UX
FR_HEAL	Player can heal in the infirmary	UR_UX
FR_ARREST	Player can arrest hostiles	UR_UX
FR_REAL_TIME	Game must run in real time (not turn based)	UR_LOGIC
FR_WIN_CONDITION	Game is won when all eight infiltrators have been arrested	UR_LOGIC
FR_LOSS_CONDITION	Game is lost when more than 15 key systems have been destroyed	UR_LOGIC
FR_TOP_VIEW	Game must be from a top down view	UR_UX
FR_MAP_SIZE	Map must have 10 rooms	UR_MAP
FR_SYSTEM_HEALTH	Key systems should have set, non-regenerating health	UR_LOGIC
FR_SYSTEM_DESTROY	Systems should be destroyed by attackers in take time depending on the difficulty	UR_LOGIC
FR_PLAYER_SPEED	Player's movement speed should be faster than infiltrators	UR_PLAYER

FR_HOSTILES_RUN	Hostiles should run from the player's teleportation gun	UR_HOSTILES
FR_ARREST	Player can't arrest hostiles if they aren't attacking anything	UR_PLAYER
FR_RESPAWN	Player should respawn at cloning bay if killed	UR_PLAYER
FR_HOSTILES_ATTACK	Hostiles' attacks should do 10% of the player's health	UR_HOSTILES
FR_HOSTILES_ABILITIES	Hostiles should have following abilities: Blinding player, player slowdown, temporary invisibility, hallucinations	UR_HOSTILES
FR_ROOM_SIZES	Rooms should be different shapes and sizes	UR_MAP
FR_HOSTILES_SPAWN	All hostiles spawn on the map from the beginning of the game	UR_HOSTILES
FR_SYSTEMS_ATTACKED	A maximum of three systems can be attacked at once	UR_LOGIC
FR_ALIENS_COUNT	There should be 24 non hostile aliens on the map	UR_MAP
FR_HOSTILES_SPECIAL	There should be four hostiles with special abilities	UR_HOSTILES
FR_PLAYER_TILES	Player should have free movement between tiles; should not snap to tiles	UR_PLAYER
FR_TELEPADS	There should be three telepads on the station	UR_MAP
FR_BRIG	The brig should appear as a room on the map	UR_MAP
FR_MINIMAP	The game should display a minimap that shows what room the player is in	UR_MAP
FR_PLAYER_VIEW	The game should be zoomed in on the player	UR_MAP
FR_TELEPAD_DESTINATION	The player can teleport from any telepad to any other telepad	UR_PLAYER
FR_PLAYER_SPAWN	Player's spawn location on the map is fixed	UR_PLAYER
FR_ALIEN_SPAWN	Spawn locations for aliens should be random	UR_MAP

FR_HOSTILES_SP AWN	Spawn locations for aliens should be random	UR_HOSTILES
FR_POWERUPS	Arresting a hostile should drop a power up that the player can use.	UR_PLAYER
FR_SAVING	The player should be able to save the game.	UR_UX
FR_DIFFICULTY_S ELECTION	The game should have a difficulty selector with multiple options.	UR_UX, UR_HOSTILES

### Non-Functional System Requirements

ID	Description	Criteria for Success
NFR_ENJOYABLE	Game should be enjoyable to play by the ENG1 cohort	Majority of subjects tested report enjoying the game
NFR_DURATION	Game should last around 5-10 minutes	Majority of games finish within this timeframe
NFR_DIFFICULTY	Game should be enjoyable and not overly difficult	Majority of subjects win the game

Requirements from the brief:

- Game must be named Auber
- Map is a space station
  - At least 4 types of rooms in the station
  - Rooms can have teleportation pads which Auber can use to teleport around the station
  - There must exist key systems in the station which can be damaged
- There are 8 hostile operatives who can sabotage key systems in the station
  - There must be at least 3 distinct special abilities within the group of infiltrators
- Sabotage attempts must be reported to the user
- The player is a constable of the station whose aim is to enforce law and order
  - Player can teleport to the infirmary to heal
  - Player can arrest the infiltrators
- Should be playable and enjoyable by the ENG1 cohort
- Game must be real time
- Two stakeholders that must accommodate:
  - Customer - trying to market and sell the game
  - UoY communications office - interested in using the game for promotional activities such as open days
- Game is won when all eight infiltrators are arrested
- Game is lost when infiltrators have destroyed more than 15 systems of the station

### Interview Requirements (**From Assessment 1**):

- Must be top down 2D game
- Designed to run on laptop/computers
- Rounds should last around 5-10 minutes
- Difficulty - enjoyable but not impossible (likely to be one time players)
- Fixed map layout (this can change if we have enough time)
- Must be at least 15 key systems
- Key systems cannot be repaired
- Player can only arrest infiltrators
- At least 2 species of aliens
- Infiltrators cannot escape prison

### Up to us to decide:

- Size of station, how many rooms  
10 rooms
- Health of key systems? Should the health of the system regenerate when not attacked?  
Set health, doesn't regenerate
- How long should it take for infiltrators to destroy the system? (10 seconds)  
10 seconds (change during testing)
- Speed of enemy and speed of player (should the player be faster)  
Player should be faster to catch enemies (certain enemies can slow player)
- Should infiltrators run from the player?  
Yes - run from teleportation gun
- Can you arrest the infiltrators if they aren't attacking anything?  
No, unless cooldown on arresting ability
- What happens if you lose all your health (lose or respawn?)  
Respawn far away on the map (cloning bay)
- How much health should the infiltrators remove per attack?  
10% of health (play around during testing)
- What are the abilities of the infiltrators? (Examples, invisibility, hallucinations, slow player, infiltrator can move quicker etc)  
Blinding player (reducing vision, obscuring map)  
Invisibility (cooldown, maybe leave trail)  
Slowing the player temporarily  
Hallucinations (create temporary copies of infiltrator)
- Can the rooms be different sizes? Multiple tile rooms  
Yes, key system one tile in room, rooms different shapes and sizes
- Will infiltrators all arrive at the same time?  
Infiltrators all at station at the start, but don't all attack at once
- Can multiple systems be attacked at once?  
Capped at 3 attacked at once (change in testing)
- How many non-infiltrator npcs should there be?  
24 (3 to every infiltrator)
- Should all infiltrators have special abilities?  
4 have abilities (balance in testing)
- Player/AI snapping to tiles?  
No snapping to tiles

- How many telepads should there be?  
(3)
- Should the brig appear as a room on the station?  
Yes, room on map (might change later)
- Minimap?
  - Shows the room the player is in, cycling images out
- Follow the player or view whole map  
Zoomed in following the player
- Can you teleport to any telepad from any other telepad?  
Teleport from any to any other teleporter
- Fixed spawn for player?  
Player is fixed spawn
- Random spawn for NPC/infiltrators?  
Random spawn

Decided Requirements:

- Tile based
  - Rooms can consist of more than one tile