

A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is a light green color. They are positioned diagonally, with the blue one in front of the green one.

# BuildingManager

Emmanuel Teferra, Garrett Wood, Adebayo  
(Hamid) Afolabi, Chenxi (Charles) Mu



# Purpose

The purpose of this brief is to articulate the methods, procedures, and requirements governing the BuildingManager app, a scheduling software used to track and regulate repairs in industrial buildings.

This document will contain information on required user inputs, program constraints, technical details, and how BuildingManager operates. While the SRS is publicly available to all, the target audience of this presentation are users, software developers, programmers, and technical advisors.



# Overview

## Product

- ❖ BuildingManager is a top-down repair scheduling and building administration app designed to speed up building repairs, organize various staff members and employees in industrial/high density buildings, and minimize wasted time due to scheduling errors and miscommunications in maintenance operations
- ❖ Used by residents, students, office workers, and employees alike, BuildingManager uses a scoring algorithm to organize maintenance tickets, identify technology outages, track maintenance progress on repairs, and even help organize DPS/EMS response teams
- ❖ BuildingManager also produces periodic records that are saved, processed, and relayed to management teams in order to better allocate resources and plan repairs

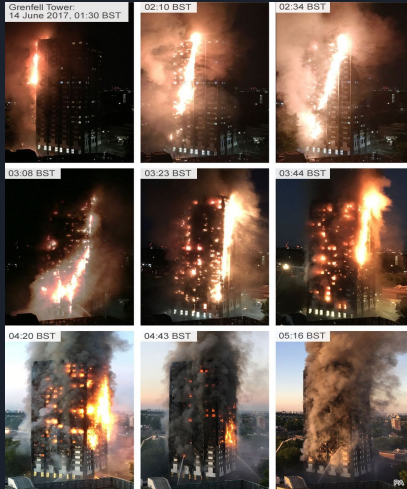


# Product Scope

- The BuildingManager app was designed for one purpose; to efficiently organize repairs in high-density buildings. With colleges, offices, and hotels returning to normalcy after the pandemic, many found that their buildings were in need of serious structural repairs.
- Previously, building issues would have to be seen, called in, and then processed by an individual employee (who creates the ticket) who then emails the report to the general maintenance staff.
  - Time is wasted
  - Maintenance tickets can be lost, ignored, or even deleted
  - Repairs follow FIFO, which ignores urgency of repair needed

# Why BuildingManager?

- Small, ignored mistakes in building repairs and electrical maintenance can snowball into bigger, costlier issues



In the case of tragic accidents like the Grenfell Tower Fire and the Surfside Condominium Collapse, overlooked building repairs and slow communication with residents saw valuable, life-saving time go wasted as rescuers struggled to help and locate panicked victims.



# Lexicon

- ❖ **Resident** - term used to designate any one outside user submitting a report, data, or repair request to the BuildingManager program. User must be apart of designated school or organization specified by client before interacting with software
- ❖ **Client** - Outside user who is employing BuildingManager; they are responsible for reframing program parameters and merging BuildingManager with their maintenance systems
- ❖ **Schedule** - the output of the BuildingManger program, the term Schedule denotes a processed tasklist created by the program that is then sent to designated locations (i.e. maintenance, ITS, )
- ❖ **Break Case** - this is a term used by BuildingManager to oversee all urgent building emergencies, injuries, and crisis; viewable by staff, client, and software developers alone



# Non-functional System requirements

- After selecting a problem, user will check their contact information and location
- Student can choose a period of time to fix their problem
- System will rank the problems based on urgency
- ITs and maintenance staff can choose their priority such as distance and skills
- Most frequent problems will have records for users



# Non-functional User requirements

R1: Based on the type of problem, the system should be able to separate it into ITS, maintenance or DPS

R2: System should also be able to rate the urgency of most inquiries

R3: There should be a process of giving the user a chance to give a review on their experience of the program





# Functional System requirements

- Database:
  - Students' address and profile
  - ITs and maintenance staff information
  - Common issues and emergencies
- User shall be able to connect with their netID
- A dropdown menu of streets and resident halls for user to select their address
- Student can enter their floor and room number, or exact location
- ITs and maintenance staff can select their operating time, office location, and their capabilities/skills
- System will acquire GPS to have staff and students' location
- A menu for user to select type to issue
- If is an emergency, system will give several contact information: DPS/police
- Non-emergent issue:
  - a dropdown menu to select problem from database
  - Select other: textbox to describe problem



# Functional User Requirements

R1: Users shall select their college which for now is Syracuse

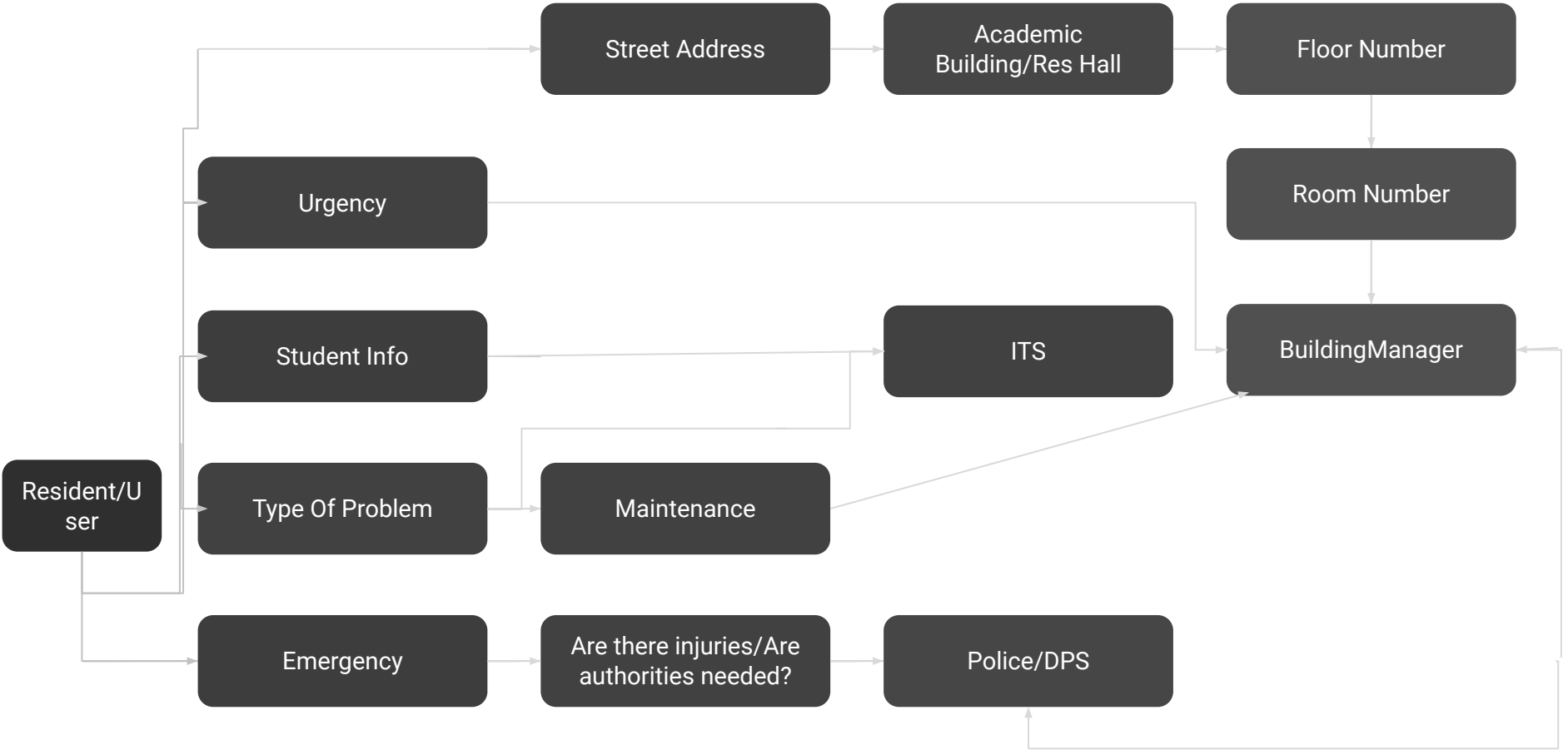
R2: User shall log in with their school info which includes their net Id and password for easy and secure verification

R3: Users will choose their exact location within the college, whether it's in one of the dorm buildings stating the specific floor and room number or a specific street and apartment/house address

R4: Users will select whether it's an emergency or pick from the drop down menu what type of problem it is

R5: Users that select emergency will then choose whether or not there were any injuries

Use Case Diagram





Thank you!