

# **RAPORT DE PROIECT**

## **Aplicație GUI pentru Planificarea Schiței de Apartament în 2D**

### **Autori:**

**Alupului Diana**

**Bran Ionuț-Alexandru**

**Găină Alexandru**

**Petrea Paul-Alberto**

**Data: 24 Octombrie 2025**

## **1. DESCRIEREA DETALIATĂ A TEMEI DE PROIECT**

### **1.1 Scopul Proiectului**

Scopul acestui proiect este dezvoltarea unei aplicații desktop interactive care permite utilizatorilor să proiecteze și să planifice amenajarea unui apartament în format 2D. Aplicația va oferi o interfață grafică intuitivă prin care utilizatorii pot crea planuri arhitecturale complete, plasând pereți, uși, ferestre și diverse elemente de mobilier într-un mod vizual și eficient.

### **1.2 Obiectivele Proiectului**

Obiectivele principale ale proiectului sunt:

- Dezvoltarea unei interfețe grafice user-friendly care să permită crearea rapidă și precisă a planurilor de apartament
- Implementarea unui sistem de grilă milimetrică configurabilă pentru plasarea exactă a elementelor arhitecturale
- Crearea unei biblioteci extinse de obiecte predefinite (pereți, uși, ferestre, mobilier) care pot fi adăugate prin simpla acțiune de drag-and-drop
- Asigurarea flexibilității prin funcționalități de editare avansate
- Implementarea unui sistem de straturi pentru organizarea eficientă a diferitelor componente ale planului
- Oferirea de instrumente de măsurare și calcul automat al suprafețelor pentru fiecare cameră
- Permitearea salvării și exportului proiectelor

### **1.3 Descrierea Aplicației Dezvoltate**

Aplicația va consta într-un mediu de lucru bazat pe canvas, similar cu aplicațiile profesionale de floor planning precum Sweet Home 3D sau Floorplanner. Zona centrală va afișa o foaie milimetrică configurabilă pe care utilizatorul va putea plasa elementele arhitecturale. Meniul lateral va conține categorii organizate de obiecte, iar panoul de proprietăți va permite ajustarea precisă a dimensiunilor, poziției și aspectului fiecărui element selectat. Utilizatorii vor putea începe fie de la zero, fie din șabloane predefinite pentru apartamente cu 2 sau 3 camere. Aplicația va include funcționalități avansate precum verificare coliziuni, calcul automat al suprafețelor, și suport pentru comentarii și adnotări pe plan.

## **2. INTERACȚIUNEA CU UTILIZATORUL**

### **2.1 Acțiunile Utilizatorului**

Utilizatorul va interacționa cu aplicația prin multiple modalități:

- Crearea structurii: Selectarea instrumentelor pentru pereți din meniul lateral și desenarea acestora pe canvas prin click și drag.
- Adăugarea ușilor și ferestrelor: Plasarea ușilor și ferestrelor prin drag-and-drop pe pereții existenți. Utilizatorul va putea ajusta direcția de deschidere a ușilor.
- Mobilarea spațiilor: Accesarea categoriilor de mobilier și plasarea obiectelor dorite prin drag-and-drop pe canvas. Fiecare obiect poate fi rotit, redimensionat și poziționat exact.

- **Editarea proprietăților:** Selectarea oricărui element de pe canvas pentru a accesa panoul de proprietăți, unde pot fi modificate dimensiunile exacte, poziția, unghiul de rotație, culoarea, materialul și alte atribute.
- **Organizarea prin straturi:** Utilizarea sistemului de layers pentru a separa structura, mobilierul și instalațiile, cu posibilitatea de a ascunde sau afișa fiecare strat independent.
- **Măsurători și calcule:** Utilizarea instrumentului de măsurare pentru a determina distanțele între puncte și accesarea calculatorului de suprafețe pentru a obține automat aria fiecărei camere.
- **Salvare și export:** Salvarea proiectului în format propriu pentru editare ulterioară sau exportul în formate standard pentru prezentare sau imprimare.
- **Navigare în canvas**

## **2.2 Feedback-ul Oferit Utilizatorului**

Aplicația va oferi feedback vizual și informațional constant:

- **Snap to grid:** Când un obiect este plasat sau mutat, acesta se va alinia automat la grila milimetrică, oferind feedback vizual prin evidențierea punctelor de aliniere.
- **Highlighting și selecție:** Obiectele vor fi evidențiate la hover și vor avea un contur colorat când sunt selectate.
- **Verificare coliziuni:** Aplicația va detecta și va semnaliza vizual suprapunerile nedorite între obiecte sau situații problematice.
- **Tooltip-uri informative:** La hover peste butoane și instrumente, vor apărea tooltip-uri care explică funcția acestora.
- **Indicatori de stare:** Bara de stare va afișa informații precum zoom-ul curent, scara utilizată, unitatea de măsură.
- **Notificări:** Mesaje de confirmare pentru acțiuni importante (salvare reușită, export finalizat) și avertismente pentru potențiale probleme.
- **Calcule automate:** Afișarea automată a suprafeței totale și a suprafeței fiecărei camere în panoul de informații.

## **3. TEHNOLOGII UTILIZATE**

### **3.1 Limbaj de Programare**

Aplicația va fi dezvoltată în Python, un limbaj de programare versatil și popular, cu un ecosistem bogat de biblioteci pentru dezvoltarea interfețelor grafice și procesarea grafică.

### **3.2 Framework și Biblioteci**

- **PyQt5/PyQt6:** Framework principal pentru crearea interfeței grafice (GUI). Qt oferă componente puternice și flexibile pentru dezvoltarea de aplicații desktop profesionale, inclusiv widget-uri avansate pentru desenare și manipulare grafică.
- **QtCore:** Modul pentru funcționalități de bază (evenimente, semnale, sloturi, timere)
- **QtWidgets:** Modul pentru widget-uri standard (butoane, meniuri, toolbars, dialoguri)
- **QtGui:** Modul pentru funcționalități grafice (desenare, culori, fonturi, imagini)
- **QPainter:** Clasă pentru desenarea pe canvas a elementelor grafice
- **QGraphicsScene/QGraphicsView:** Pentru gestionarea scenei grafice și a interacțiunii cu obiectele

- **JSON:** Bibliotecă standard Python pentru salvarea și încărcarea proiectelor în format JSON
- **ReportLab sau QPrinter:** Pentru exportul în format PDF
- **Pillow (PIL):** Pentru manipularea și exportul imaginilor în format PNG
- **svgwrite:** Pentru exportul în format SVG

### 3.3 Mediu de Dezvoltare

**Visual Studio Code (VS Code)** va fi utilizat ca IDE principal, oferind:

- Suport excelent pentru Python prin extensii dedicate
- Debugging avansat cu breakpoint-uri și inspecție variabile
- Integrare cu Git pentru controlul versiunilor
- Terminal integrat pentru rularea și testarea aplicației
- IntelliSense pentru autocompletare și sugestii de cod
- Extensii pentru Qt Designer (pentru design vizual al interfețelor)

### 3.4 Instrumente Auxiliare

- **Qt Designer:** Instrument vizual pentru proiectarea rapidă a interfețelor grafice
- **Git/GitHub:** Pentru controlul versiunilor și colaborarea în echipă

## 4. COMPONENTA ECHIPEI ȘI DISTRIBUȚIA SARCINILOR

**Alupului Diana:**

**Rol:** Responsabil cu arhitectura generală a aplicației și implementarea logicii de bază.

**Sarcini:**

- Proiectarea arhitecturii aplicației și a structurii claselor
- Implementarea sistemului de gestionare a proiectelor (creare, deschidere, salvare)
- Dezvoltarea modelului de date pentru reprezentarea obiectelor (pereți, uși, ferestre, mobilier)
- Implementarea sistemului de coordonate și grilă milimetrică
- Dezvoltarea funcționalităților de transformare (rotire, redimensionare, poziționare)
- Integrarea tuturor componentelor dezvoltate de ceilalți membri

**Bran Ionuț-Alexandru:**

**Rol:** Responsabil cu interfața grafică și experiența utilizatorului.

**Sarcini:**

- Proiectarea și implementarea interfeței grafice principale (meniu superior, toolbar, meniu lateral)
- Dezvoltarea paginii de start cu opțiuni pentru proiect nou, deschidere și șabloane
- Implementarea panoului de proprietăți pentru editarea obiectelor selectate
- Crearea sistemului de teme (Day/Night mode)
- Implementarea dialog-urilor și ferestrelor modale
- Asigurarea consistenței vizuale și a design-ului intuitiv
- Crearea iconițelor și a elementelor vizuale

**Găină Alexandru:**

**Rol:** Responsabil cu canvas-ul principal și funcționalitățile de desenare.

**Sarcini:**

- Implementarea canvas-ului principal cu QPainter și QGraphicsScene
- Dezvoltarea sistemului de snap to grid și aliniere automată
- Implementarea funcționalităților de zoom, pan și rotire canvas
- Dezvoltarea instrumentelor de măsurare și calculare suprafețe
- Implementarea sistemului de layers (straturi) pentru organizarea elementelor
- Dezvoltarea funcționalității de drag-and-drop pentru plasarea obiectelor
- Implementarea verificării coliziunilor și validărilor
- Crearea sistemului de selecție și manipulare vizuală a obiectelor

**Petrea Paul-Alberto:**

**Rol:** Responsabil cu biblioteca de obiecte și funcționalitățile de import/export.

**Sarcini:**

- Crearea bibliotecii complete de obiecte predefinite (pereți, uși, ferestre, mobilier)
- Implementarea categoriilor de mobilier (dormitor, living, bucătărie, baie, birou, diverse)
- Implementarea funcționalității de export proiect
- Dezvoltarea sistemului de șabloane predefinite (apartamente 2/3 camere)
- Crearea sistemului de ajutor și tutorial interactiv
- Documentarea codului și crearea ghidului de utilizare

Fiecare membru va colabora strâns cu ceilalți prin întâlniri regulate de sincronizare, utilizând Git pentru controlul versiunilor și integrarea continuă a modificărilor. Responsabilitățile pot fi ajustate în funcție de progresul proiectului și de necesitățile specifice care apar în timpul dezvoltării.