Graphical user interface

Description automatically generatedYEOJU TECHNICAL INSTITUTE

IN TASHKENT

Amaliy Informatika kafedrasi

**O’quv amaliyot hisoboti**

Mavzu: “Java dasturlash tilining JavaFX kutubxonasi yordamida savol-javob desktop ilovasi”

**Guruh:** ISE-31U

**Bajardi:** Fozilov Rustam

**Amaliyot rahbari:** Xudoyberdiyev Azimbek

Toshkent – 2022

**MUNDARIJA**

|  |  |  |
| --- | --- | --- |
| **I.** | **KIRISH. LOYIHA QISQACHA TASNIFI** | 3 |
| **II** | **MAVJUD TIZIMLAR TAHLILI, REJALASHTIRISH VA LOYIHALASH** | 4-17 |
| 2.1. | O’rganilgan dasturlar tahlili va Brainstorming natijalari | 4-7 |
| 2.2. | Ishlab chiqilgan funksional talablar. Loyiha bo’yicha ishlab chiqilgan talablar jadvali va ularning to’liq tasnifi | 8-9 |
| 2.3. | To-be model. Loyiha arxitekturasi: flowchart, sequence diagram. Loyihaning modullari va ularning tasnifi. | 10-11 |
| 2.4. | Dastur interfeysini loyihalash. | 12-14 |
| 2.5. | Ma’lumotlar bazasi modelini loyihalash | 15-17 |
| **III** | **DASTUR ISHLAB CHIQISH** | 18-21 |
| 3.1. | JavaFX da ishlab chiqilgan dastur oynalari (fxml ko’rinishlari va tarkibiy tuzilishi) | 18-21 |
| 3.2. | JavaFX CSS orqali ishlatilgan qo’shimcha dizayn tasnifi | 22 |
| 3.3. | Dastur modullari ishlash jarayoni (Dastur modullarini yaratish uchun ishlatilgan Controllerlar, Application sinfi va ularning o’zaro bog’lanishi) | 23-24 |
| 3.4. | Dastur ma’lumotlarini boshqarish (*CRUD[create, update, delete, edit] qismi*) | 25-27 |
| 3.5. | Foydalanilgan instrumentalvositalar | 28 |
| **IV** | **XULOSA.** | 29 |
| **V** | **FOYDALANILGAN ADABIYOTLAR** | 30 |
| **VI** | **ILOVALAR** | 31 |

**I. KIRISH. LOYIHANING QISQACHA TASNIFI**

**Loyihaning dolzarbliligi.** Mazkur loyiha foydalanuvchilarga yangi o’rgangan ko’nikmalarini tekshirib ko’rishlari uchun, test qilib ko’rishlari uchun ularga qulay sharoit yaratib beradi.

* Bilimlarni tekshirib, mustahkamlab borish imkoniyatini beradi.
* Ushbu dastur har qanday foydalanuvchiga ushbu ilovadan foydalanish imkoniyatini beradi.

**Loyihani maqsadi:** Bu loyihani yaratishdan maqsad insonlarga o’z bilim darajalarini mustahkamlash va ko’rop yodda saqlab qolish imkoniyatlarini yaratish.

**II. TAHLIL QILISH, REJALASHTIRISH VA LOYIHALASH**

**2.1. O’rganilgan dasturlar va Brainstorming natijalari**

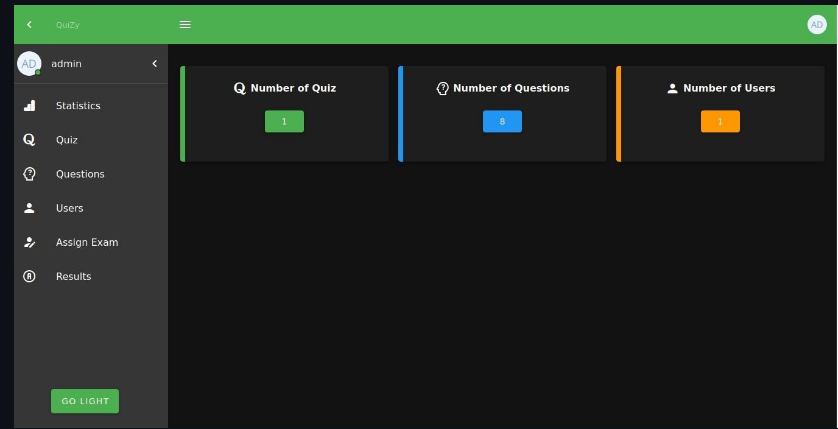
**Brainstorming:** Brainstorming (aqliy hujum) – bu g’oyalarni ishlab chiqish uchun guruhli fikrlash usuli hisoblanadi. Guruh tomonidan quyidagi fikrlar va g’oyalar bildirildi.

* Dastur qulay va sodda interfeysga ega bo’lishi kerak.
* Foydalanuvchi test ma’lumotlarini o’zgartira olishi kerak.
* “Sozlamalar” bo’limi bo’lishi kerak va shu bo’lim orqali foydalanuvchi test ma’lumotlarini o’zgartira olishi kerak.
* “Dastur haqida” bo’limi bo’lishi kerak.
* Dasturdan qanday foydalanish haqida yo’riqnoma dasturning “Dastur haqida” bo’limida yoritilishi kerak.
* Test tugagandan so’ng foydalanuvchi natijasini ekranda chop qilish kerak.
* Test tugagandan so’ng foydalanuvchining natijasini foiz ko’rsatkichida ham ko’rsatib berish kerak.
* Foydalanuvchi yangi fanlarni qo’sha olishi va o’chira olishi kerak.
* Test variantlari to’rttadan oshmasligi kerak.

**Mavzuga oid mavjud dasturlar tahlili**

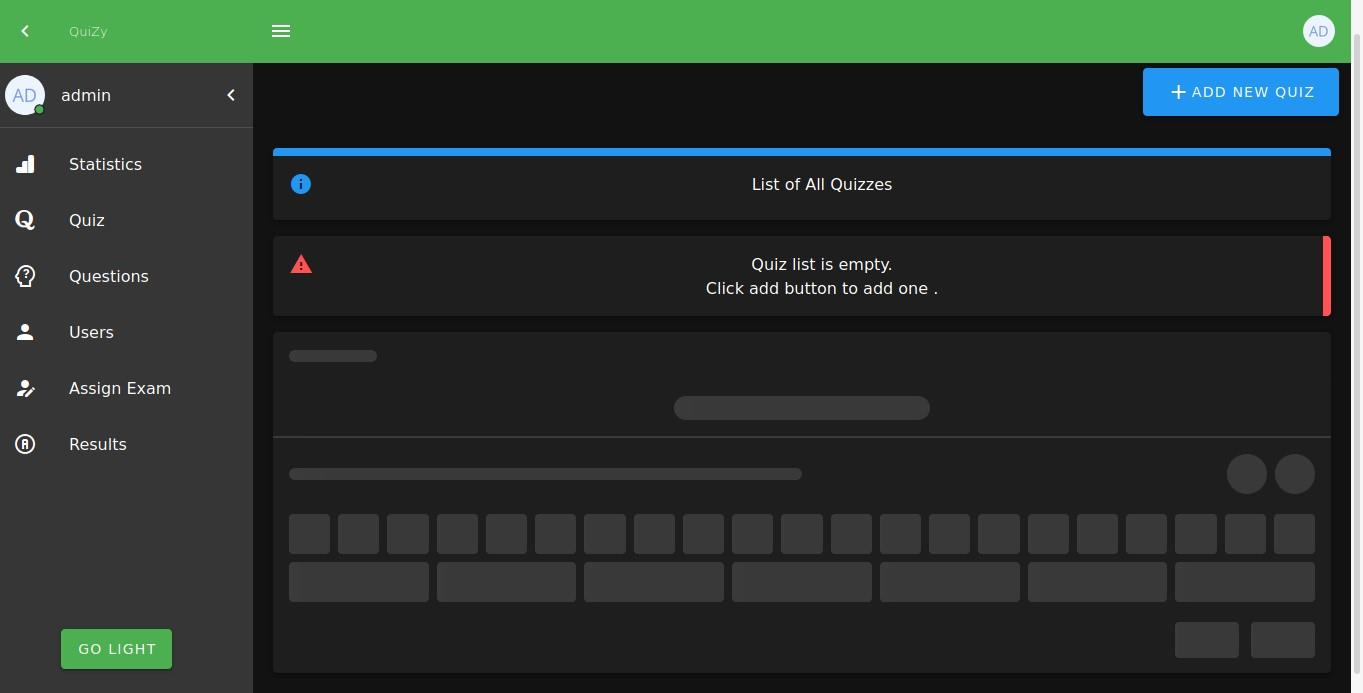
**1 - dastur.**

**Quizy**



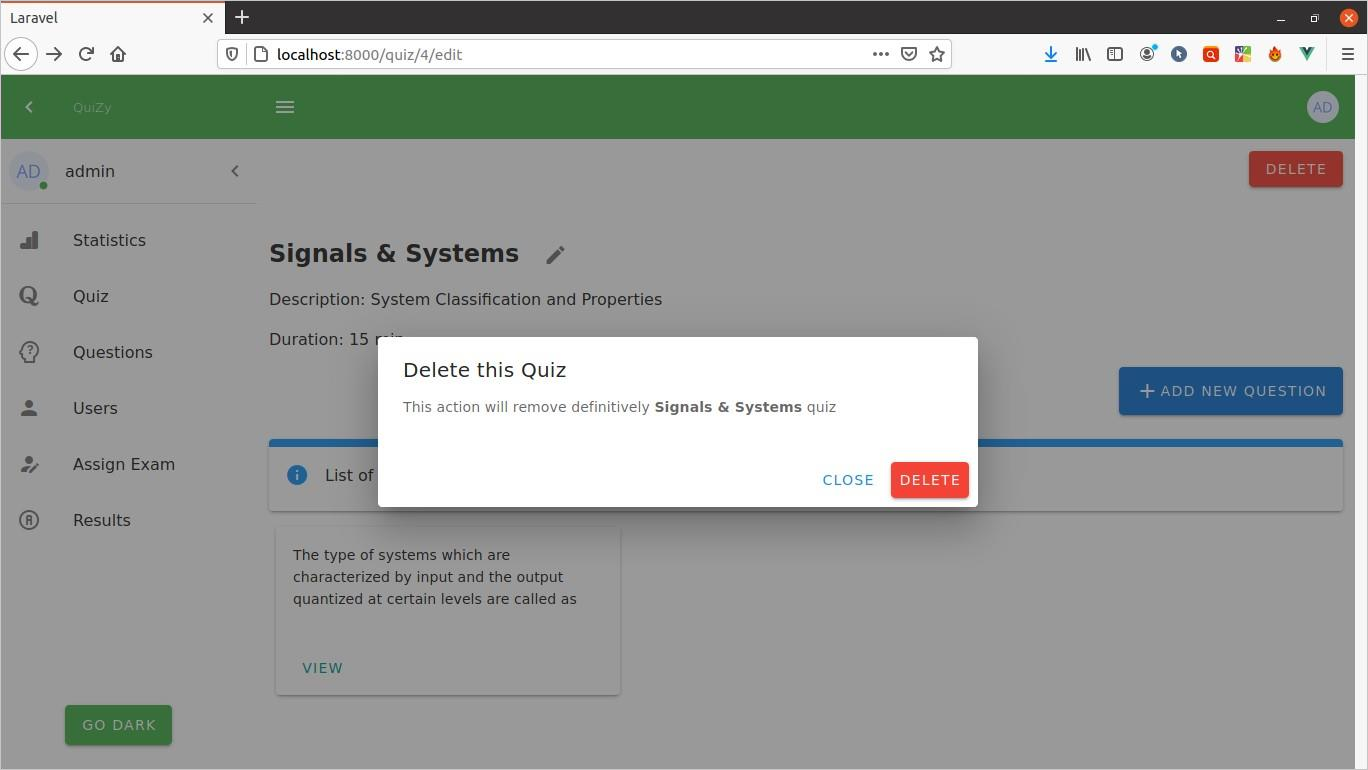
**1 - rasm**. Quizy ilovasi bosh sahifasi

Ushbu loyiha foydalanuvchilarga testlarni yechib qanday natija olganliklarini ko’rsatub beruvchi veb dastur. Loyiha PHP dasturlash tilining Laravel8 frameworki da tuzilgan.



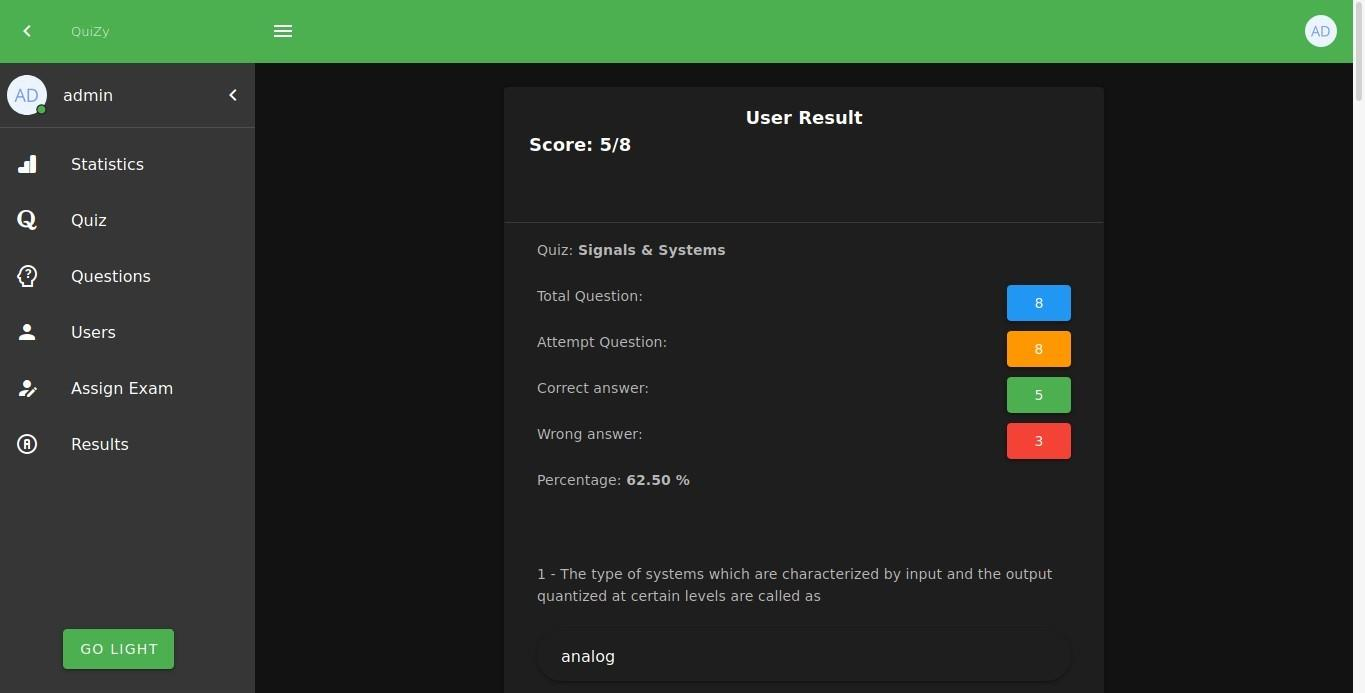
**2 - rasm.** Testlar ro’yhatini ko’rish

Bu yerda mavjud testlar ro’yhatini ko’rish mumkin. Test ham qo’shish mumkin



**3 - rasm.** Testni o’chirish oynasi

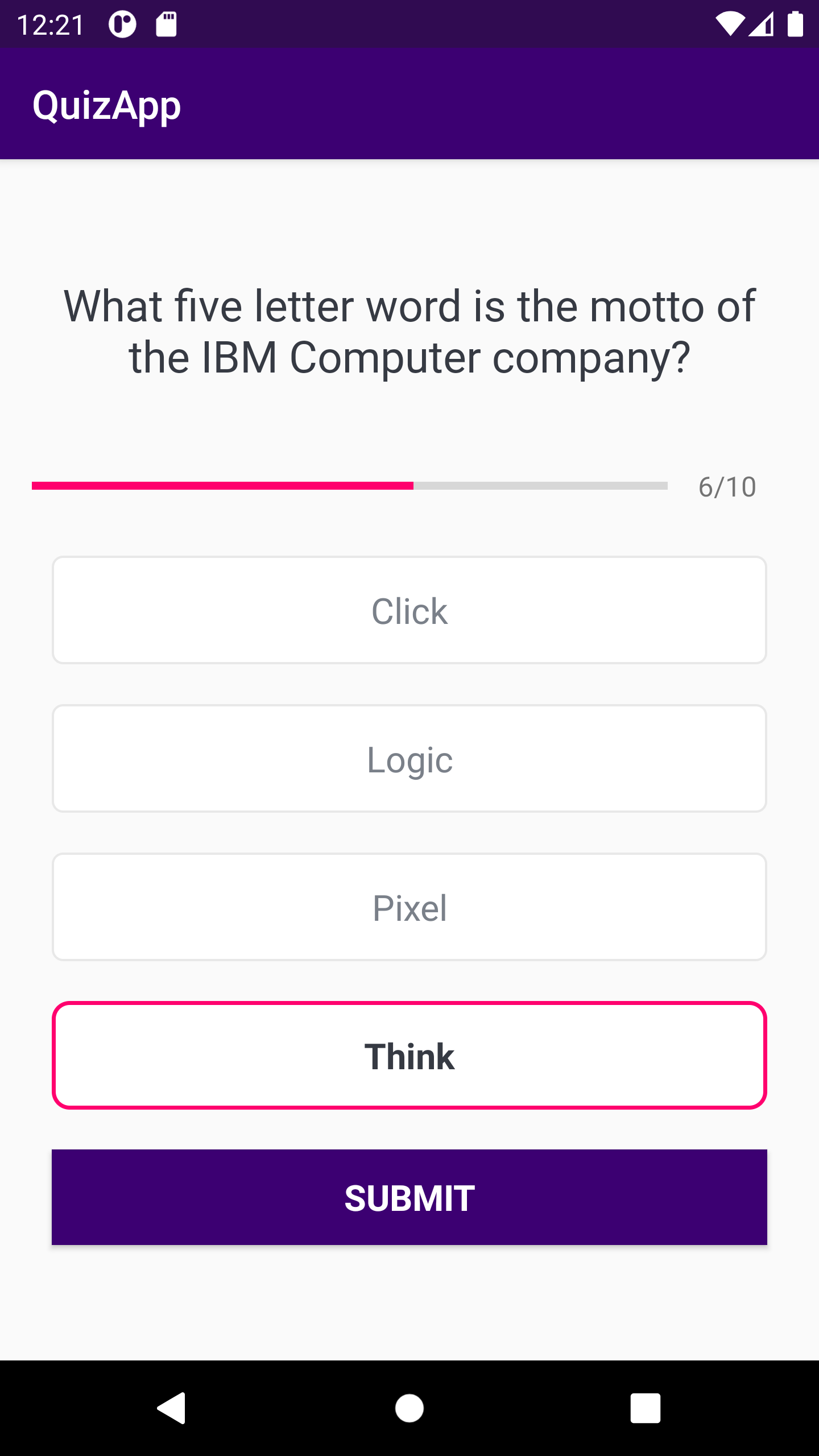
Bu yerda mavjud fanni o’chirib tashlash imkoniyati ko’rsatilgan.



**4 - rasm.** Natija oynasi

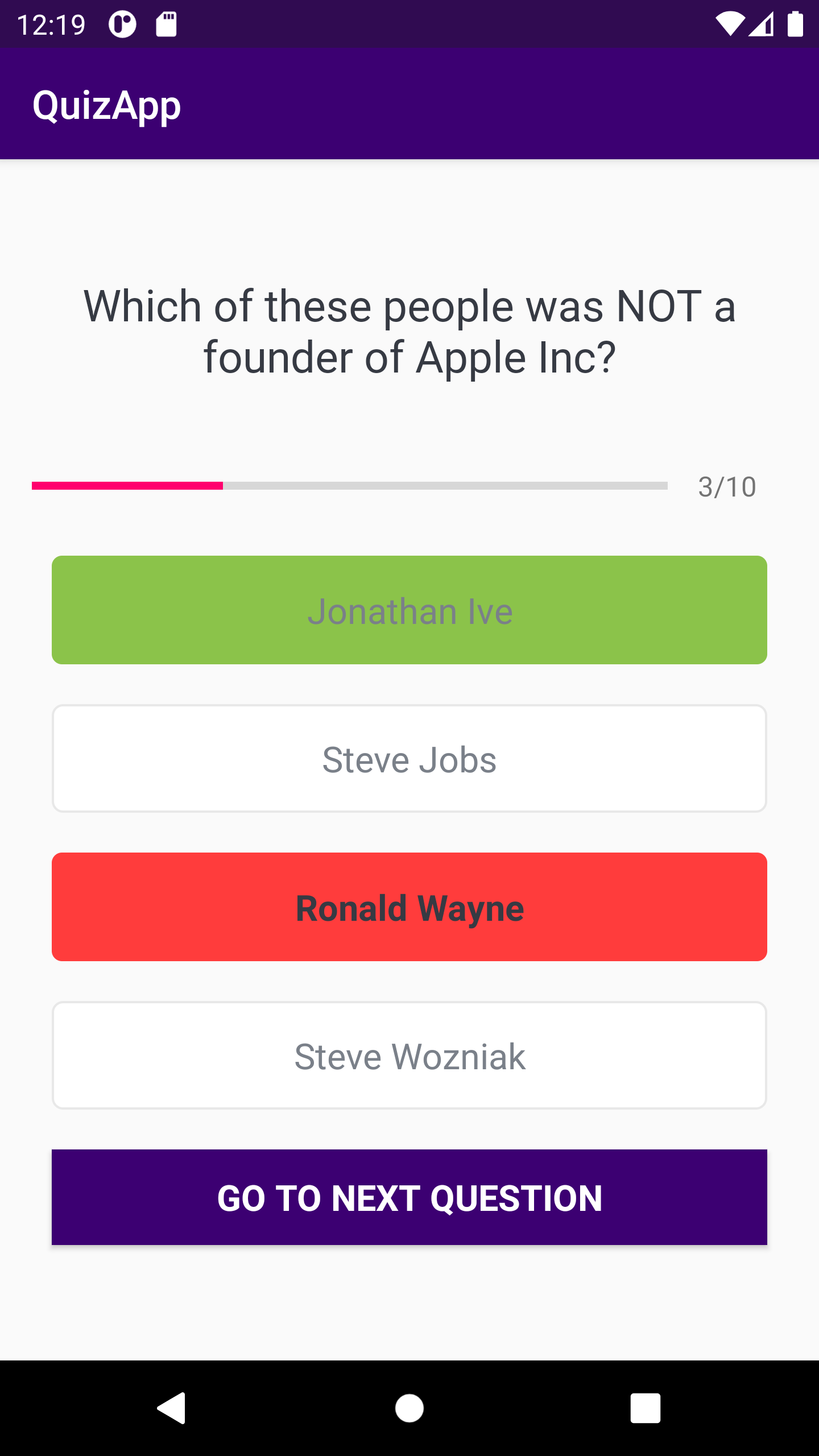
Bu rasmda foydalanuvchining natijalari va foiz ko’rsatkichlari batafsil tarzda aks etgan. Dastur manbasi: <https://github.com/Etienne-bobo/quiz-App>

**2 – dastur: QuizApp**



**5 - rasm.** QuizApp ilovasi

QuizApp mobil ilovasi. Bu dasturda ham foydalanuvchilar o’z bilimlarini sinash maqsadida test yechib ko’rishlari mumkin bo’ladi. Bu dastur faqat android foydalanuvchilari uchun.



**6 - rasm.** Test yechish jarayoni

Bu yerda dastur foydalanuvchiga test tugamasdan oldin xatoni bilish imkoniyatini ko’rsatmoqda. Dastur manbasi: <https://github.com/Solideizer/QuizApp>

**2.2. Ishlab chiqilgan funksional talablar. Loyiha bo’yicha ishlab chiqilgan talablar jadvali va ularning to’liq tasnifi**

**Loyiha talablarining turlari**

**Foydalanuvchi talablari** - bu tizimning **tabiiy tildagi batafsil tavsifi** bo’lib, uning tadqim etadigan **xizmatlar diagrammasi va operatsion cheklovlarini** tasniflaydi. Bunday turdagi talablar foydalanuvchilar tomonidan ishlab chiqiladi.

**Tizim talablari** - Tizimning barcha funksiyalari, xizmatlari va operatsion cheklovlari haqidagi tavsiflarni batafsil bayon etuvchi aniq va strukturalangan hujjat. Tizim talablari asosan 2 turga bo’linadi:

* + Funksional talablar
  + No-funksional talablar

**Funksional talablar** - bu tizim funksionalligiga tegishli bo’lgan talablarni o’z ichiga olib, tizimning ma'lum kirishlarga qanday munosabatda bo'lishi va muayyan vaziyatlarda qanday ishlashi kerakligini tasvirlaydi. Misol uchun tizimda **ro’yhatdan o’tish ketma-ketligi** qanday, ma’lumotlar bazaga qanday ko’rinishda yoziladi, foydalanuvchi **tizimdan o’chirish** qanday amalga oshiriladi, va shu kabi talablar funksional talablar hisoblanadi.

**No-funksional talablar** - bu talablar tizimga tegishli bo’lgan attributlar tavsifini belgilaydi va to’g’ridan to’g’ri tizimning funksional jihatiga bog’liq bo’lmaydi. Bu turdagi talablarga dasturni (tizimni) ishlab chiqish **narxi**, uning **samaradorligi, xavfsizligi, sifati** va boshqa shunga o’xshash talablar kiradi.

Tizimning funksional talablari va tasnifi.

**1-jadval. Funksional talablar jadvali**

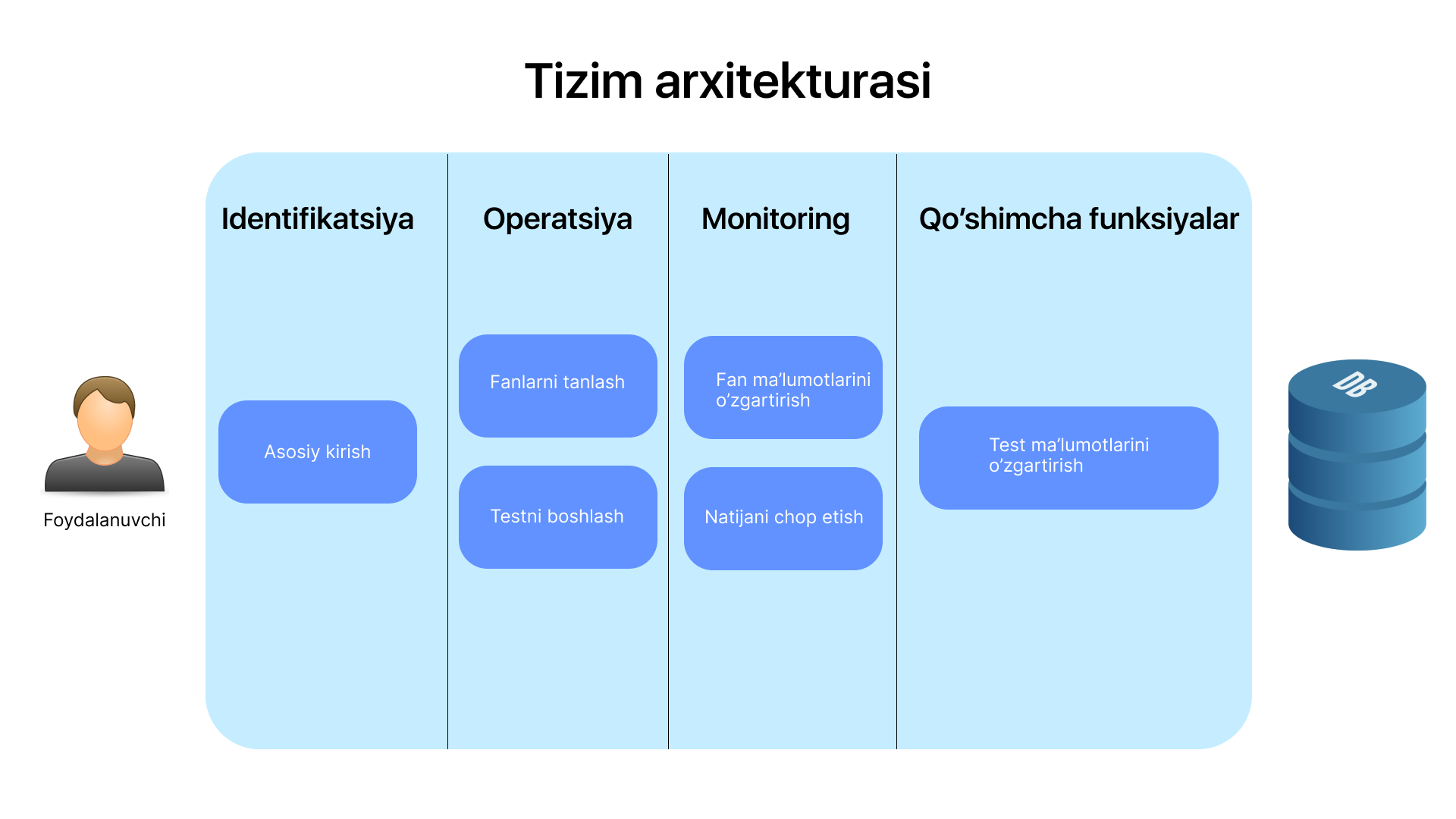
|  |  |  |
| --- | --- | --- |
| **Talab**  **identifikatori** | **Talab tasnifi** | **Talab**  **Prioriteti** |
| **Req001** | Dastur qulay va oson interfeysga ega bo’lishi kerak | **3** |
| **Req002** | Dastur ishga tushishi bilan foydalanuvchilarga mavjud testlarni ko’rsatib beruvchi oyna ochilishi kerak. | **3** |
| **Req003** | Foydalanuvchi ma’lum bir fanni tanlaganda shu fan haqida ma’lumot chiqishi kerak. | **3** |
| **Req004** | Foydalanuvchi fanni tanlab “boshlash” tugmasini bosganda test boshlanishi va shunga mos oynaga o’tishi kerak | **3** |
| **Req005** | Test variantlari 4tadan kam yoki ko’p bo’lmasligi kerak. | **3** |
| **Req006** | Foydalanuvchi testni yakunlagandan so’ng uning natijalari (to’g’ri javoblar soni) sonli va foizli ko’rinishda yangi oynada chop qilinishi kerak. | **3** |
| **Req007** | Foydalanuvchi testlar ustida CRUD (yaratish, ko’rish, yangilash, o’chirish) amallarini bajara olishi kerak. | **3** |
| **Req008** | Foydalanuvchi har bir test uchun qilgan yangilanishini ma’lumotlar bazasida saqlanib borilishi kerak. | **3** |
| **Req009** | Testlarni yangilash yoki ular ustida CRUD amallarini bajarish chog’ida kerakli maydonlar bo’sh bo’lsa “barcha maydonlar to’ldirilishi shart” degan habar chiqishi kerak. | **3** |
| **Req010** | Dasturda fanlarni o’chirish yoki qo’shish amallari bo’lishi kerak | **3** |
| **Req011** | Yangi test yoki fan qo’shilganda “Muvaffaqiyatli qo’shildi” degan habar chiqishi lozim. | **3** |
| **Req012** | Test yechish jarayonida “Testni tugatish” tugmasi bosilganda test tugab, foydalanuvchiga uning natijalari chop etilishi kerak | **3** |
| **Req013** | Test variantlari joylashuvi random tarzda joylashishi kerak. | **3** |
| **Req014** | Dasturning “Dastur haqida” bo’limiga o’tilganda dastur haqida batafsil ma’lumot va uning yo’riqnomasi ko’rsatilishi kerak. | **3** |
| **Req015** | Sozlamalar menyusida fanlar ustida CRUD amallarini bajarib beruvchi menyular bo’lishi kerak. | **3** |
| **Req016** | Sozlamalar bo’limidagi “fanni o’chirish” menyusiga o’tilgan “Chindan ham shu fanni o’chirmoqchimisiz” habari va yonida checkbox button bo’lishi kerak. | **3** |
| **Req017** | Fanni o’chirish menyusidagi checkbox button belgilangandagina fanni o’chirish amali bajarilshi va ma’lumotlar bazasidan ham o’chirilishi kerak. | **3** |
| **Req018** | Mavjud fanlar nomlari bilan nomlangan buttonlar joylashuvi TilePane qolipi ichida joylashgan bo’lishi kerak. | **3** |
| **Req019** | Yangi fan qo’shilganda shu fanga mos button yaratilishi va TilePane ichiga joylashishi kerak. | **3** |
| **Req020** | Yangi fan qo’shilgan navbatning ohiriga qo’lishi va unda so’ng “+ Test qo’shish” tugmasi keyingi navbatga o’tishi kerak. | **3** |

**2.3. To-be model. Loyihaning arxitekturasi. Loyihaning modullari va ularning tasnifi**

**To-Be model:**

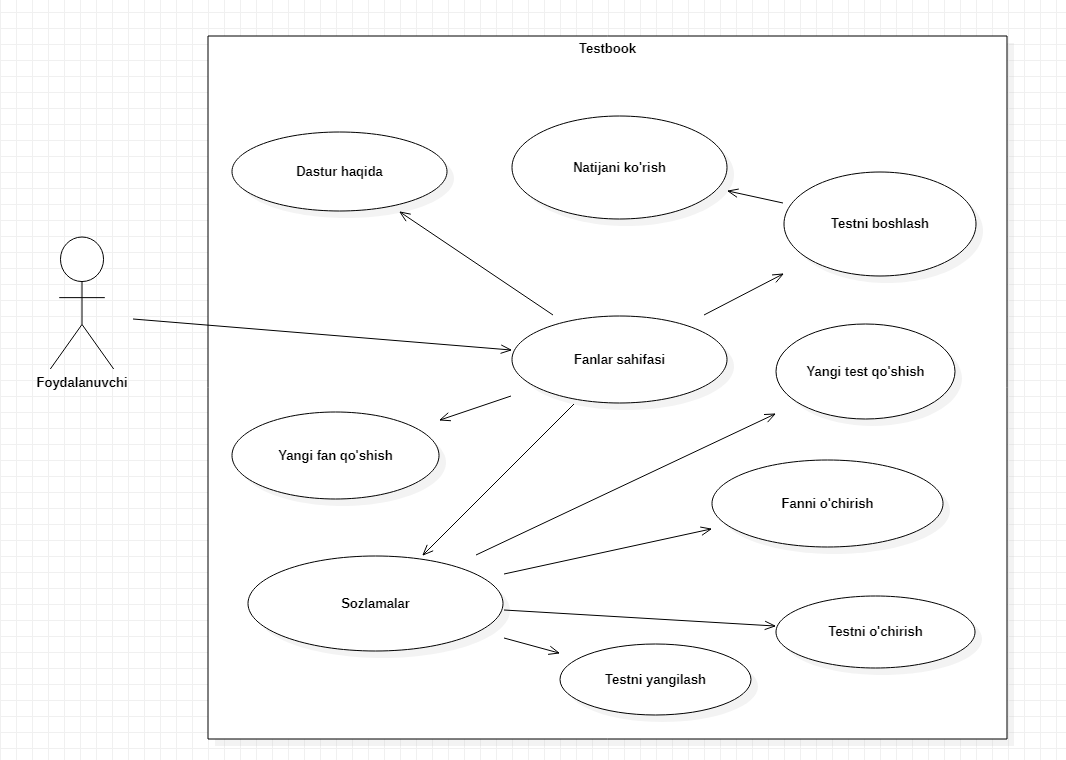
* Yangi o’rgangan ko’nikmalarni mustahkamlab borish
* Test ma’lumotlarini istalgancha o’zgartirish
* Ma’lumotlarni faqatgina bazadan olib ishlatish
* Qulay va oson interfeysga ega bo’lish

Quyidagi rasmda tizimning umumiy ko’rinishini ifodalovchi tizim arxitekturasi keltirilgan. Ushbu arxitekturada tizim foydalanuvchilari, tizim modullari va ma’lumotlarni saqlash modeli (fayl yoki relatsion ma’lumotlar bazasi) ko’rsatiladi va ularning mantiqiy bir-biriga bog’liqligi namoyish qilinadi.



**7 - rasm.** Tizim arxitekturasi.

Foydalanuvchi va ma’lumotlar bazasi o’rtasida bog’lanishlar.



**8 - rasm.** Loyihaning use-case diagrammasi.

Foydalanuvchi va uning strukturaviy bog’lanish jadvali.

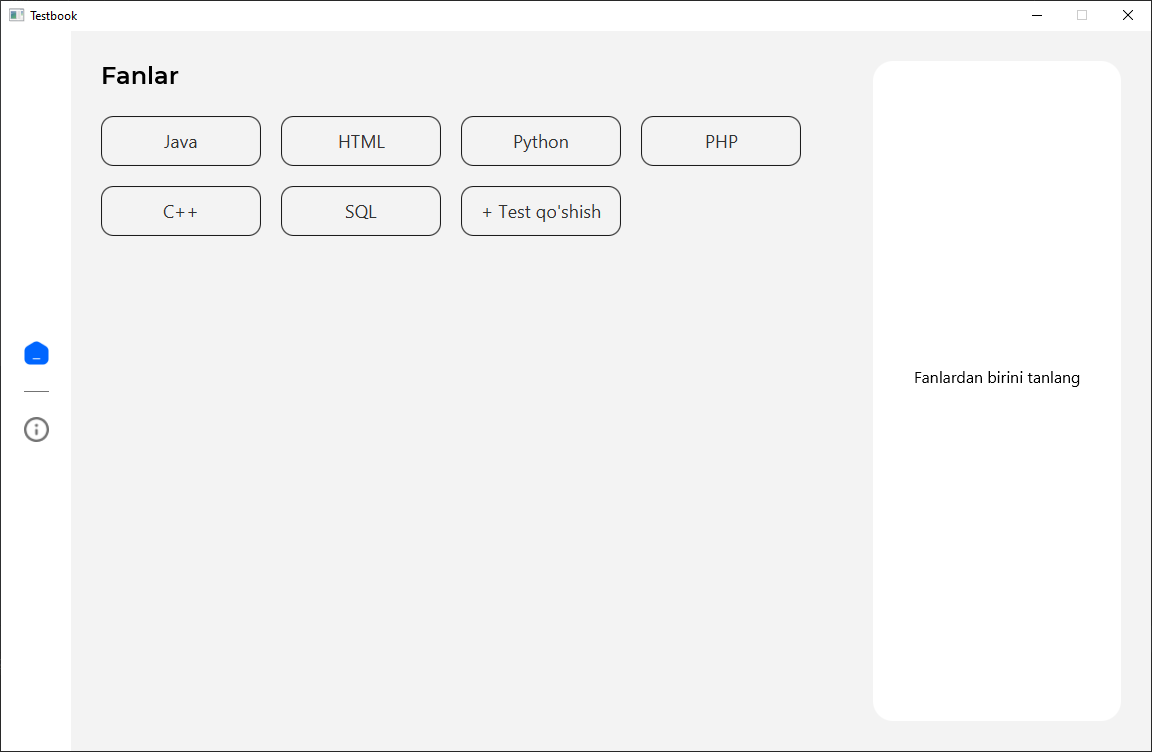
**Tizim Foydalanuvchilari:**

* Barcha turdagi foydalanuvchilar uchun.

**Tizim modullari:**

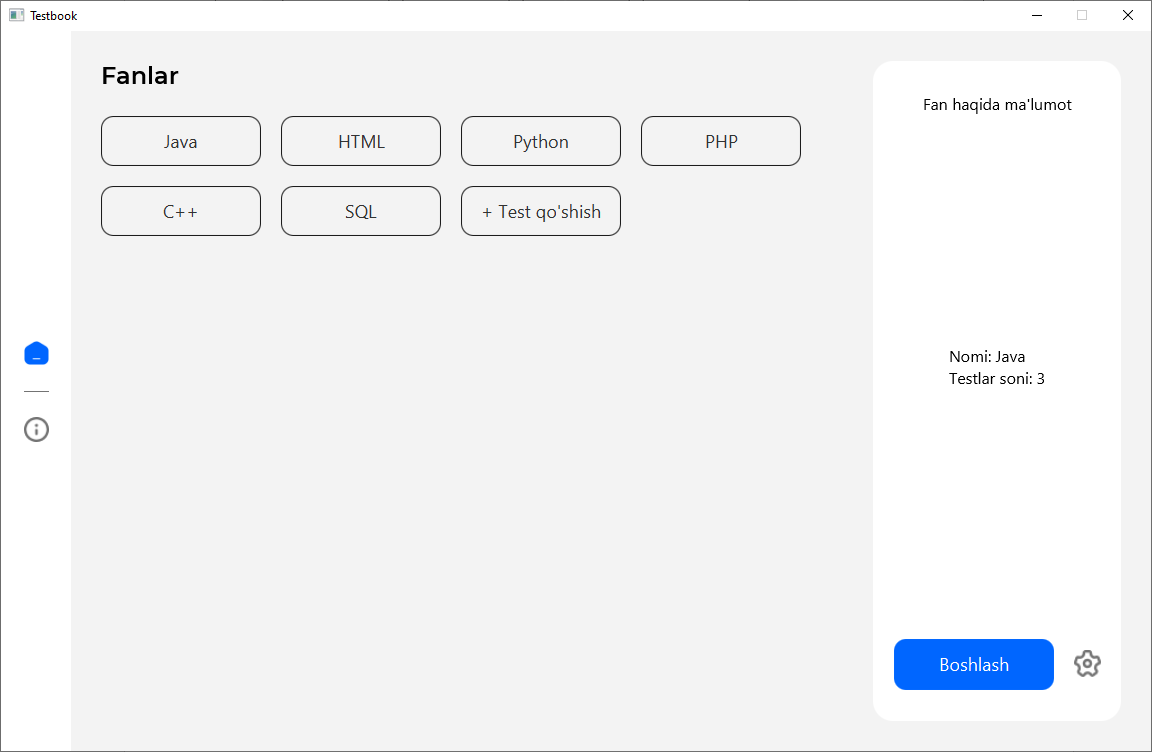
* + Fanlar moduli.
  + Test yechish moduli
  + Natija moduli.
  + Sozlamalar moduli.
  + Dastur haqida moduli

**2.4. Dastur interfeysini loyihalash**



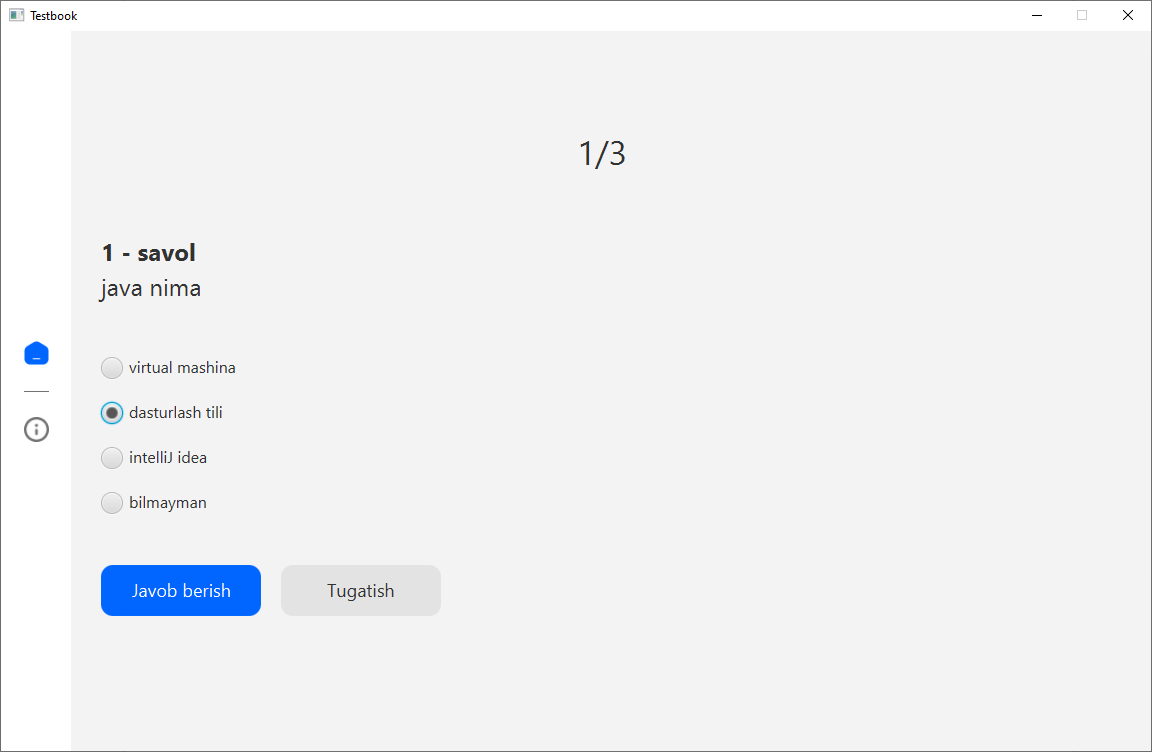
**9 - rasm.** Dasturning bosh sahifasi.

Bosh sahifada mavjud fanlar nomlari bilan nomlangan tugmalar va “+ Test qo’shish” tugmasi mavjud.



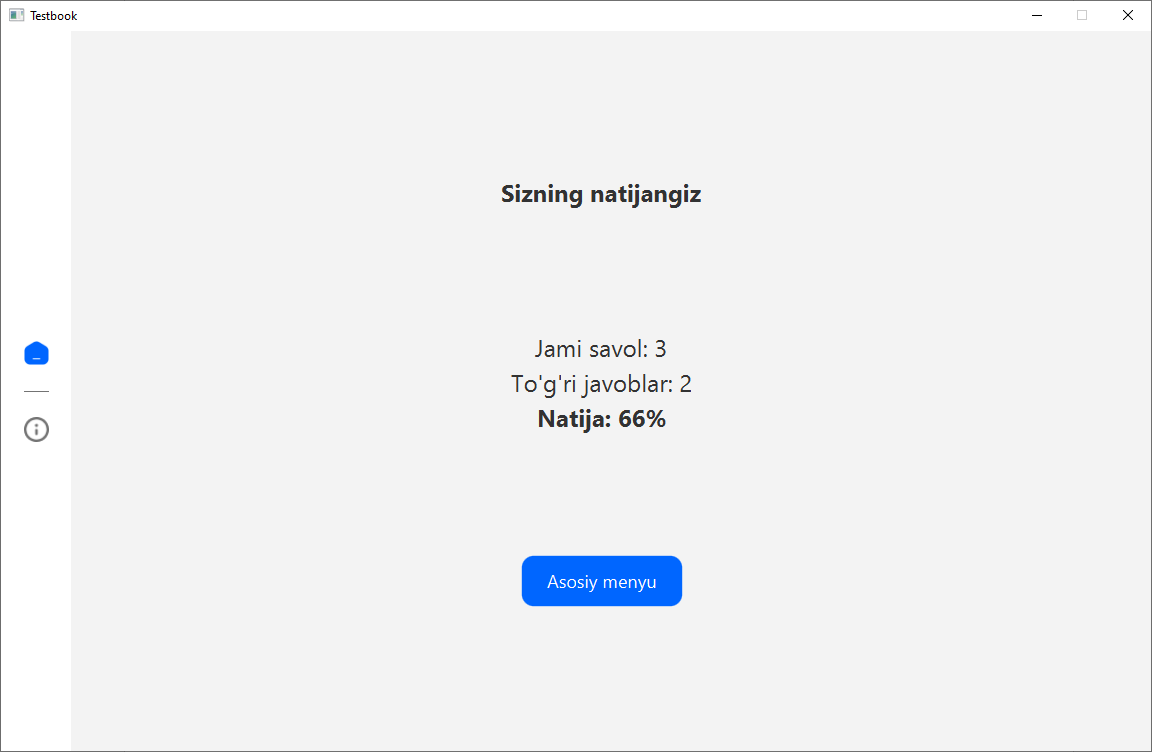
**10 - rasm.** Fan haqida ma’lumot oynasi.

Bu yerda tanlangan fan haqida ma’lumot ko’rsatiladi.



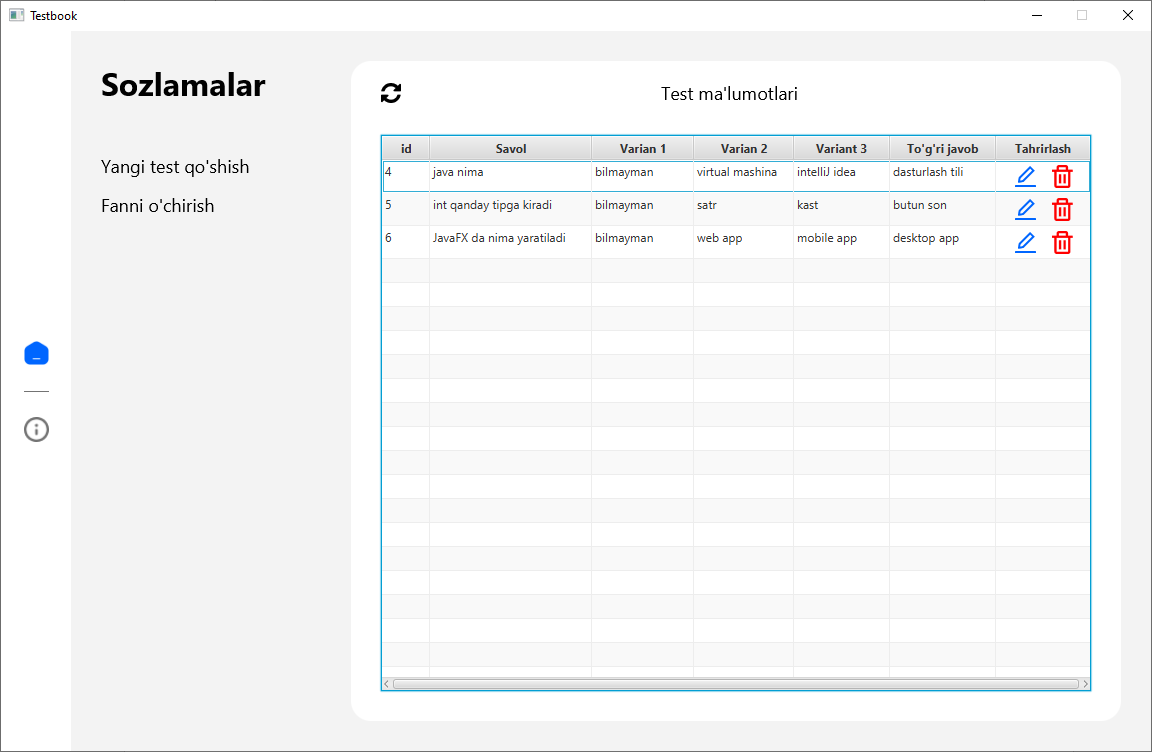
**11 - rasm.** Test boshlanish sahifasi.

Fanga biriktirilgan testlar va uning variantlari ma’lumoti chop etiladi



**12 - rasm.** Natija oynasi.

Bu yerda siz yechgan testlaringizning natijasini fozi va sonli ko’rinishda ko’rishingiz mumkin.



**13 - rasm**. Sozlamalar oynasi.

Fan haqida ma’lumotlar va ular ustida amallar bajarish uchun “sozlamalar” oynasi ko’rinishi.

**2.5. Ma’lumotlar bazasi modelini loyihalash**

**Ma’lumotlar bazasi konseptual modeli** – bu tahlil qilingan va ishlab chiqilgan talablar asosida ma’lumotlar bazasini tashkil etuvchi elementlari (jadvallar,so’rovlar) va ularning mohiyat-bog’lanish modelini qurishdan iborat. MB konseptual modeli Top-down va Bottom-up ko’rinishlarda amalga oshiriladi. Konseptual modelning top-down tahlil qilish usulida ma’lumotlar bazasiga qo’yilgan talablar asosida MB elementlari shakllantiriladi. Konseptual modelning bottom-up tahlil usulida esa ishlab chiqilishi kerak bo’lgan tizimga tegishli ma’lumotlar asosida MB elementlarini shakllantiriladi. Axborot tizimining ma’lumotlar bazasi konseptual modeli quyidagi qismlardan iborat bo’ladi:

- loyiha haqida, uning maqsadi va vazifalari;

- ma’lumotlar va ma’lumotlar bazasiga qo’yiladigan umumiy va funksional talablar;

- ma’lumotlar bazasini tashkil qiluvchi jadvallar va ularning to’liq tasnifi;

- jadvallarni shakllantirish (extracting entities);

- jadval atributlarini shakllantirish (extracting attributes and determining ID keys)

- jadvallar orasidagi mohiyat-bo’glanish (Entity-Relationship) modeli;

- normallashtirish formalarini joriy qilish.

**Konseptual model tashkil etuvchilari tasnifi**

**Loyiha nomi:** “Testbook” ko’nikmalarni mustahkamlab borish uchun savol-javob ilovasi

**Loyiha haqida, uning maqsadi va vazifalari.** Ushbu qismda ishlab chiqilayotgan ma’lumotlar bazasiga mos axborot tizimi yoki dasturiy vositaning ahamiyati, ma’lumotlar bazasini tashkil qilish ko’lami (scope), ma’lumotlar bazasining asosiy maqsadi va vazifalari keltiriladi.

**Ma’lumotlar va ma’lumotlar bazasiga qo’yiladigan umumiy va funksional talablar.** Konseptual modelni qurishning mazkur bosqichida tanlangan obyektga tegishli ma’lumotlar tahlil qilinadi, ma’lumotlar modeli shakllantiriladi hamda ma’lumotlar bazasini tashkil qilishdagi umumiy va funksional talablar keltiriladi. Bunda ma’lumotlar bazasida qanday turdagi ma’lumotlar saqlanishi, konseptual model qurishning qaysi usulidan foydalanilishi, ma’lumotlar bazasidan foydalanuvchilar kategoriyalari va boshqa shu kabi umumiy va funksional talablar shakllantiriladi.(2-jadval)

**2-jadval. Ma’lumotlar bazasiga qo’yiladigan talablar**

|  |  |  |
| --- | --- | --- |
| **Talab**  **identifikatori** | **Talab turi** | **Talab tasnifi** |
| **T001** | UT | Ma’lumotlar bazasi relatsion ma’lumotlar bazasi modeli asosida qurilishi lozim |
| **T002** | UT | Ma’lumotlarni saqlashda fayllar tizimidan foydalanish nazarda tutilmaydi |
| **T003** | UT | Konseptual model yaratishda “top-down” tahlil  qilish usulidan foydalanish lozim |
| **T004** | UT | Ma’lumotlar bazasini har haftada nusxasini  (backup) ko’chirib olish lozim |
| **T006** | FT | Tizimga tegishli ma’lumotlarni saqlashda fayllar tizimidan foydalanish imkoniyati bo’lishi lozim |
| **T007** | FT | Bitta fanga bir nechta test biriktirilishi mumkin. |
| **T008** | FT | Fanlar id si test ma’lumotlari bilan bo’glangan bo’lishi kerak. |

Qisqartmalar: UT – umumiy talab, FT – funksional talab

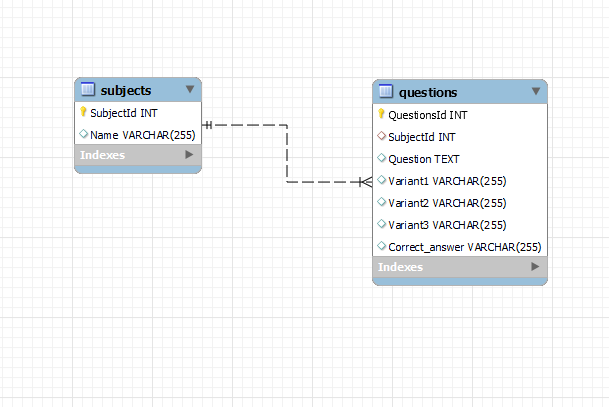
**Ma’lumotlar bazasini tashkil qiluvchi jadvallar va ularning to’liq tasnifi.**

Mazkur bosqichda ma’lumotlar bazasini tashkil qiluvchi jadvallar shakllantiriladi (3-jadval). Ishlab chiqilgan talablar asosida har jadval bo’yicha atributlar va kalit ustunlar shakllantiriladi .

**3-jadval. MB dagi jadvallar ro’yhati**

|  |  |  |
| --- | --- | --- |
| **#** | **Jadval nomi** | **Jadval tasnifi** |
| **1** | Subjects | Fan ma’lumotlari saqlanadigan jadval. |
| **2** | Question | Fanlarga biriktirilgan test ma’lumotlari saqlanadigan jadval. |

**Database bog’lanishlari**



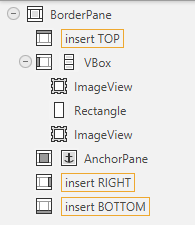
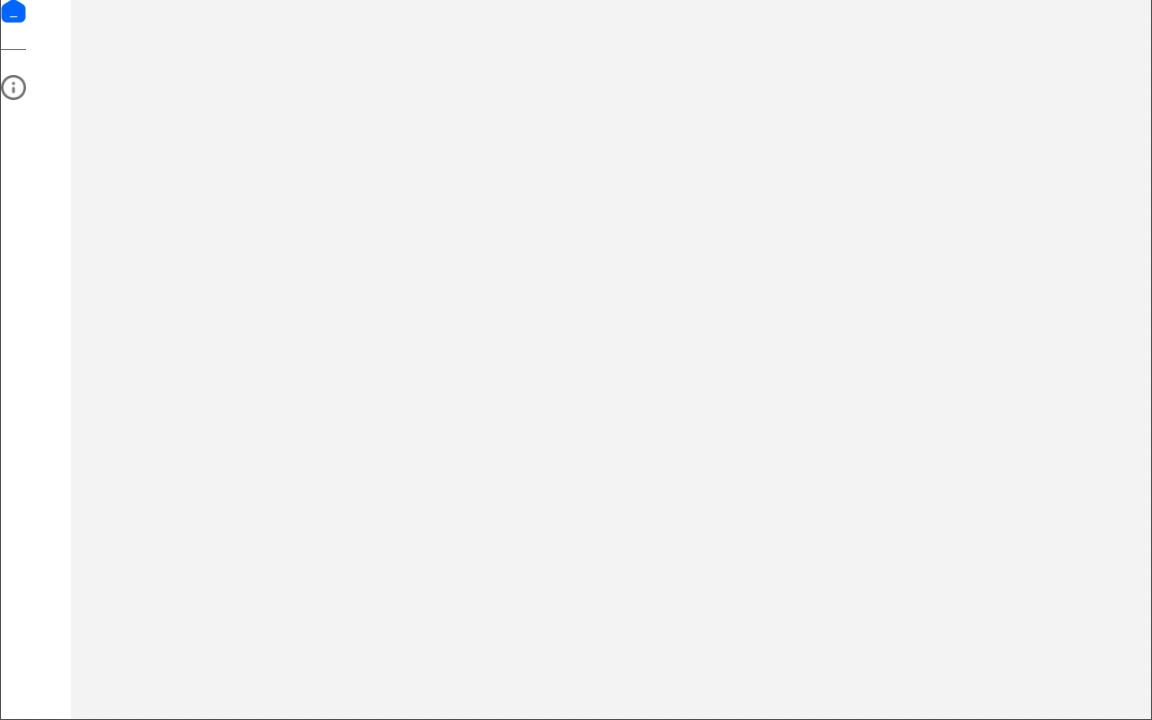
**14 - rasm**. Ma’lumotlar bazasi sxemasi**.**

Bu sxemada ikkita jadval va ularning ierarxik bog’lanishlar tasvirlangan.

**III DASTUR ISHLAB CHIQISH**

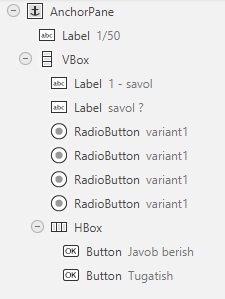
**3.1. JavaFX da ishlab chiqilgan dastur oynalari (fxml ko’rinishlari va tarkibiy tuzilishi)**

**Umumiy fxmllar.**



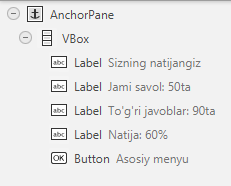
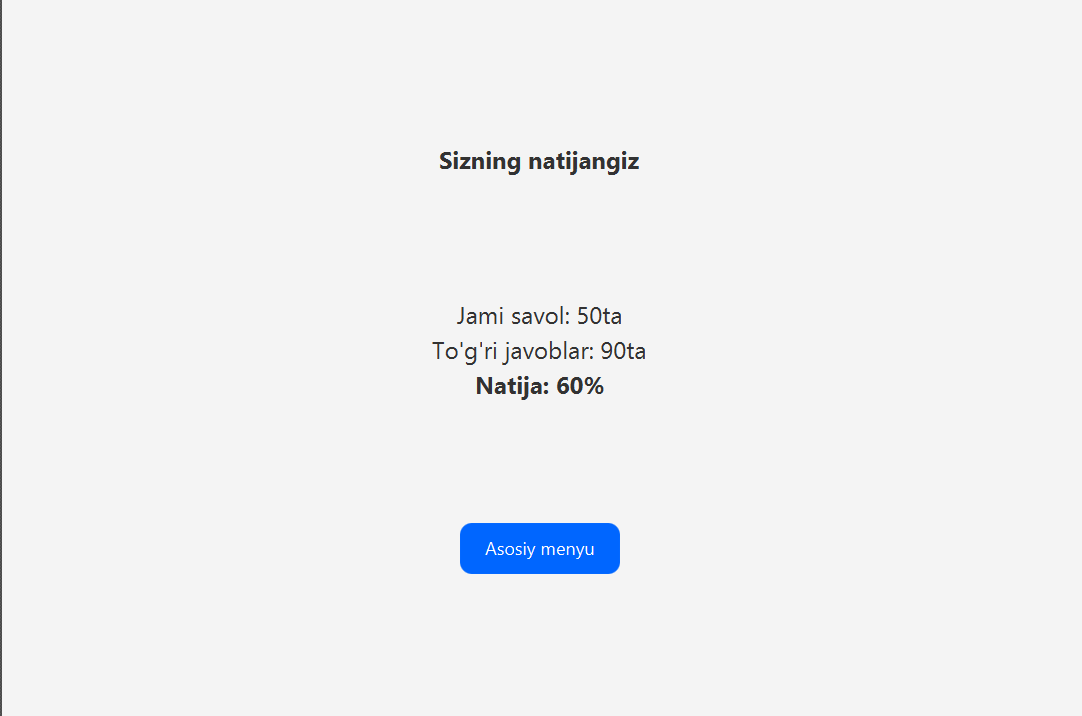
**15 - rasm.** Hello-view.fxml ko’rinishi va tarkibiy tuzilishi.

Dastur ishga tushgan payti ko’rinuvchi oyna.



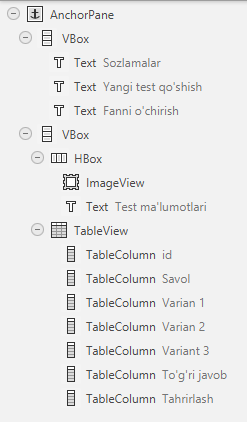
**16 - rasm.** Test-started.fxml ko’rinishi va tarkibiy tuzilishi.

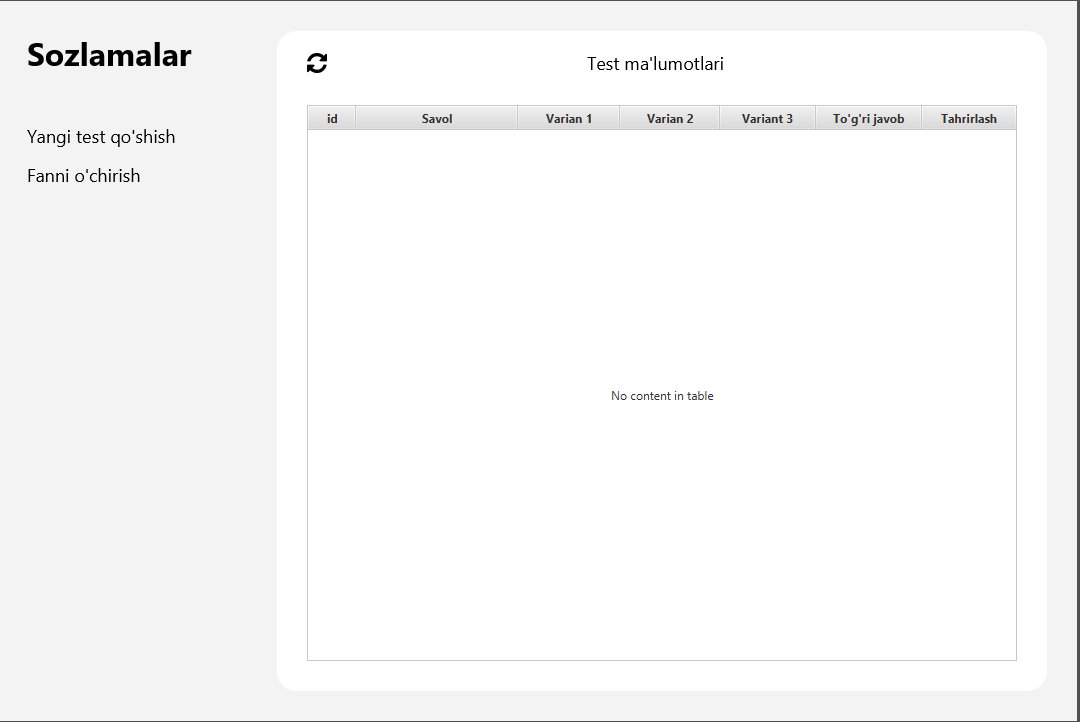
Savol boshlanish oynasidagi test va uning varinatlari.



**17 - rasm.** Result.fxml ko’rinishi va tarkibiy tuzilishi.

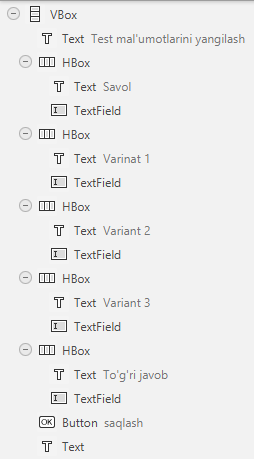
Test yechib bo’lingandan keying natija oynasi (foiz va sonli ko’rinishda).

****



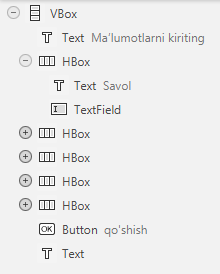
**18 - rasm**. Settings.fxml ko’rinishi va tarkibiy tuzilishi.

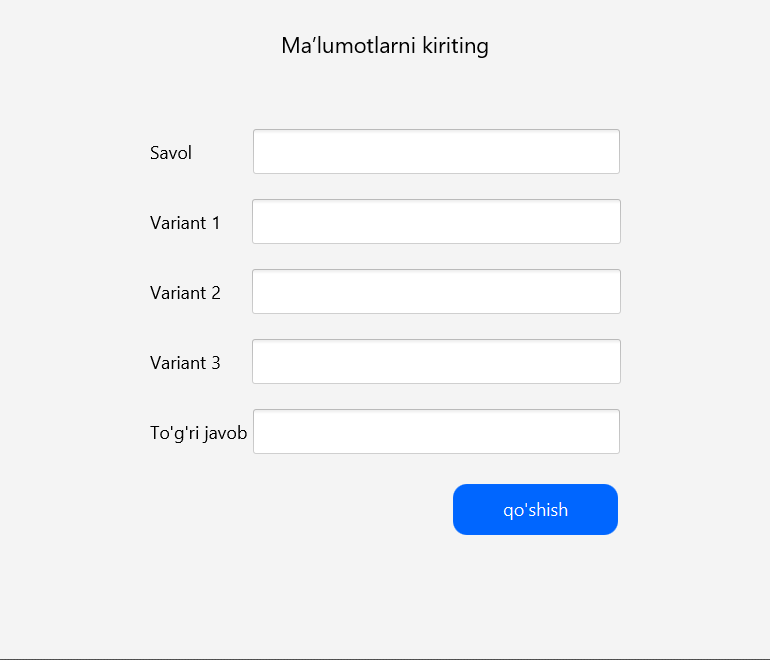
Fanlarning sozlamalar bo’limi va uning tuzilishidagi strukturalar.



**Rasm19.** Edit-test.fxml ko’rinishi va tarkibiy tuzilishi.

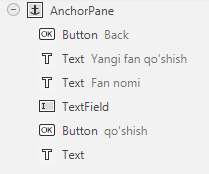
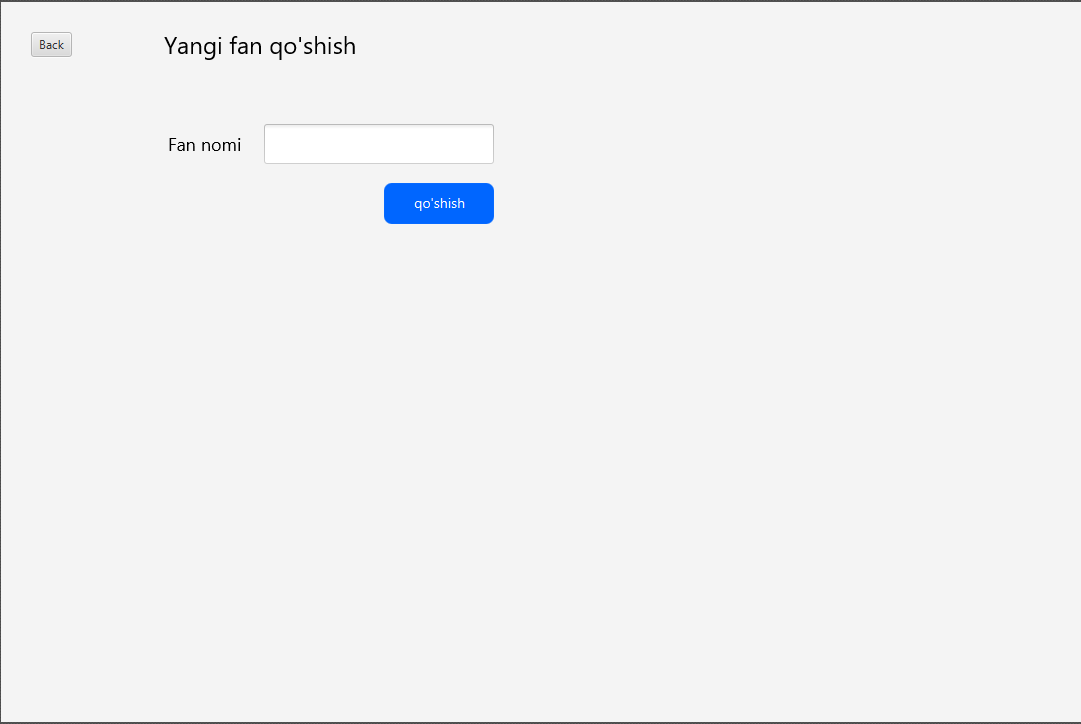
Test ma’lumotlarini yangilash va saqlash imkoniyati ko’rsatilgan.

****



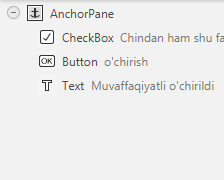
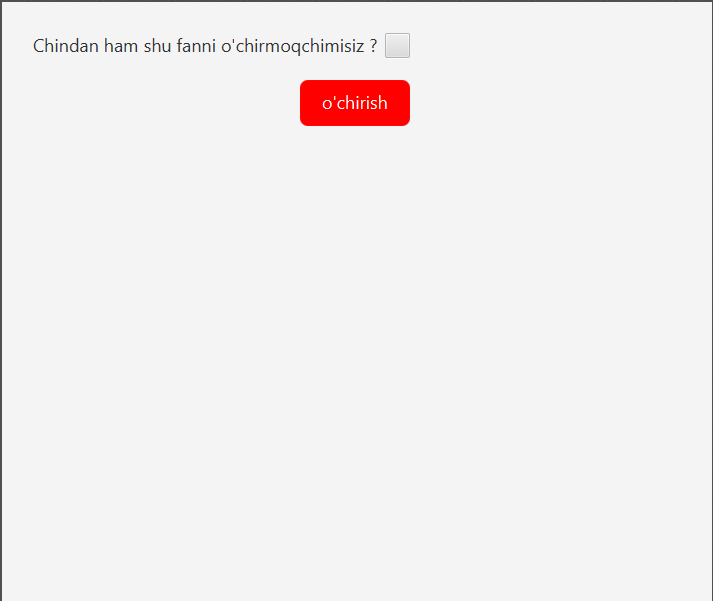
**20 - rasm.** Add-test.fxml ko’rinishi va tarkibiy tuzilishi.

Yangi test qo’shish oynasining strukturasi.



**21 - rasm.** Add-subject.fxml ko’rinishi va tarkibiy tuzilishi.

Yangi fan qo’shish oynasida fan nomini kiritish kifoya.



**22 - rasm.** Delete-subject.fxml ko’rinishi va tarkibiy tuzilishi.

Belgini belgilash orqali tanlangan fan o’chirilib yuboriladi.

**3.2 JavaFX CSS orqali ishlatilgan qo’shimcha dizayn tasnifi**

**Asosiy sahifaning dizayni uchun ishlatilgan css elementlari**

**Css elementlarning vazifasi**

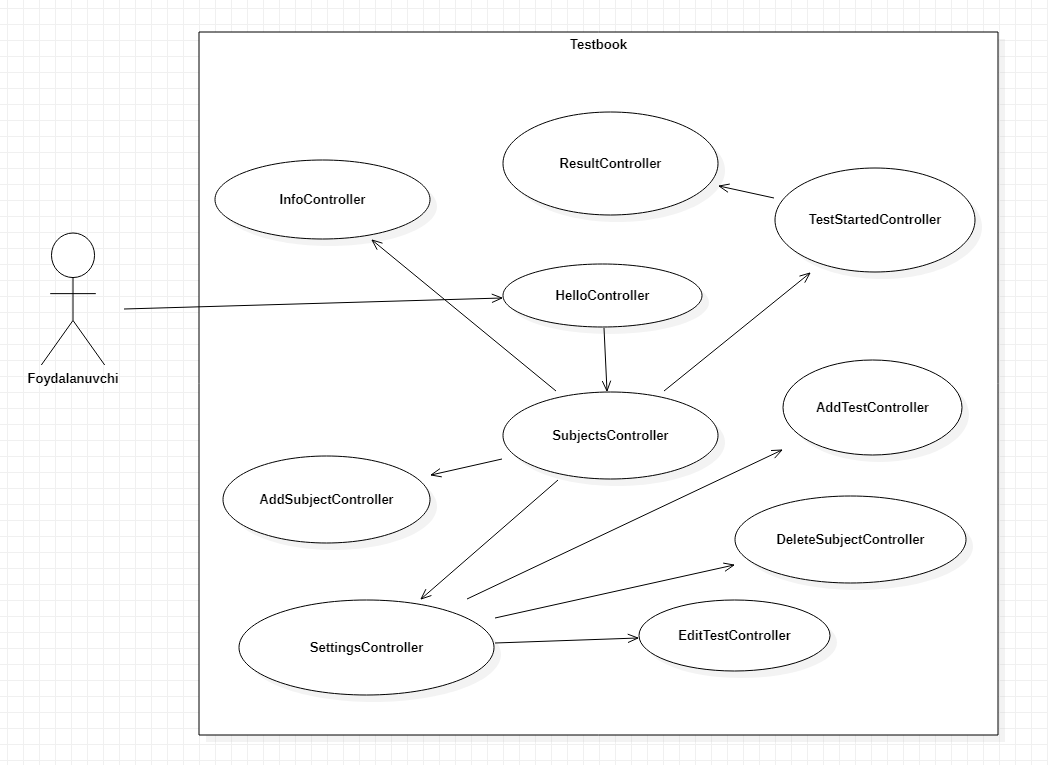
Dastur sarlavhasi uchun yangi shrift o’rnatish.

Kerakli belgilar ustiga sichqoncha orqali borilganda (hover) belgining shakli o’zgarishi.

Tugmalarga yangi shrift o’rnatish va shrift o’lchamini berish.

@font-face {  
 font-family: 'test';  
 src: url('Fonts/Montserrat-SemiBold.ttf');  
}  
  
@font-face {  
 font-family: 'test2';  
 src: url('Fonts/Montserrat-SemiBold.ttf');  
}  
  
#homeIcon:hover {  
 -fx-image: url("../../../hover-home-icon.png");  
}  
  
#homeIcon {  
 -fx-image: url("../../../hover-home-icon.png");  
}  
  
#aboutIcon {  
 -fx-image: url("../../../about-icon.png");  
}  
  
#title {  
 -fx-font-family: "Montserrat SemiBold";  
}  
  
#addTestButton {  
 -fx-font-family: "Montserrat SemiBold";  
 -fx-font-size: 18px;  
}  
  
.button {  
 -fx-font-family: "Montserrat SemiBold";  
 -fx-font-size: 16px;  
}

**3.3 Dastur modullari ishlash jarayoni (Dastur modullarini yaratish uchun ishlatilgan Controllerlar, Application sinfi va ularning o’zaro bog’lanishi)**



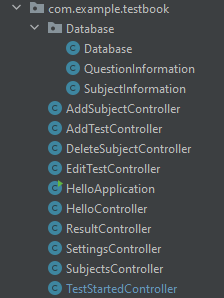
**23 - rasm.** Controllerlar orasidagi bog’lanishlar.

Jami 10ta controller va ularning bog’lanishlari diagrammasi.

**Umimiy Controllerlar**

* HelloController
* SubjectsController
* TestStartedController
* ResultController
* SettingsController
* AddSubjectController
* AddTestController
* DeleteSubjectController
* EditController

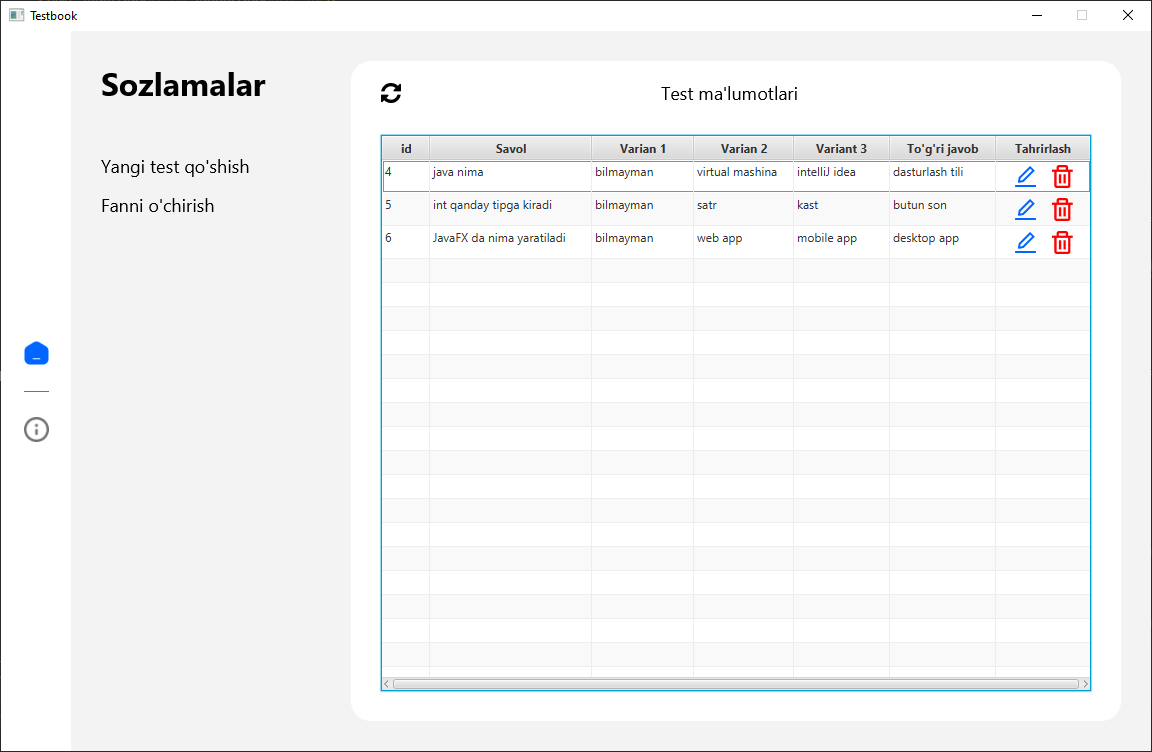
**Loyihaning umumiy strukturasi**



**24 - rasm.** Loyihaning umumiy strukturasi.

Ishlatilgan kerakli classlar jamlanmasi.

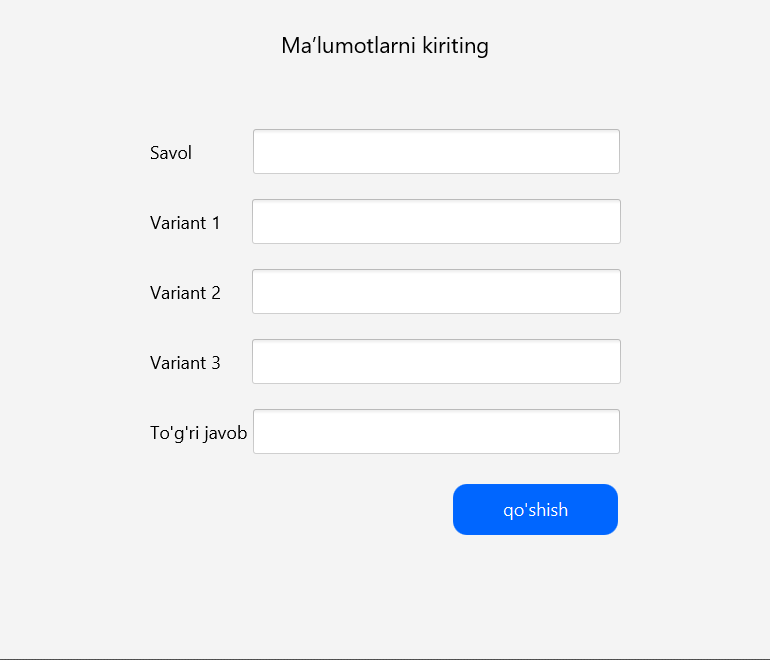
**3.4** **Dastur ma’lumotlarini boshqarish (*CRUD[create, update, delete, edit]qismi*)**



**25 - rasm.** Crud oynasi.

Bu yerda fan ustida istalgan CRUD amalini bajarish mumkin.

**Yangi test qo’shish**

****

**Rasm26.** Test qo’shish oynasi.

Bu yerda kerakli ma’lumotlar kiritiladi va qo’shish tugmasi bosiladi va test ma’lumoti qo’shiladi.

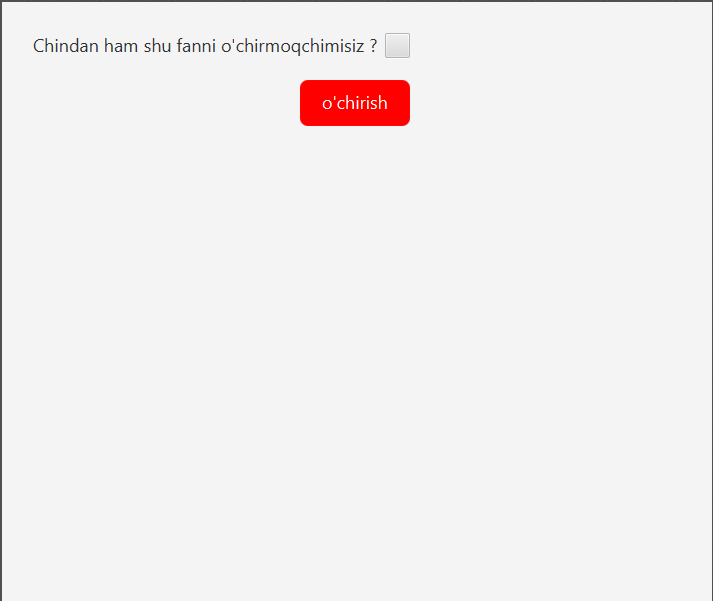
**Test ma’lumotlarini yangilash**

****

**27 - rasm.** Ma’lumotlarni yangilash oynasi.

Test uchun yangi ma’lumotlar kiritiladi va saqlash tugmasi bosilganda ma’lumotlar yangilanadi.

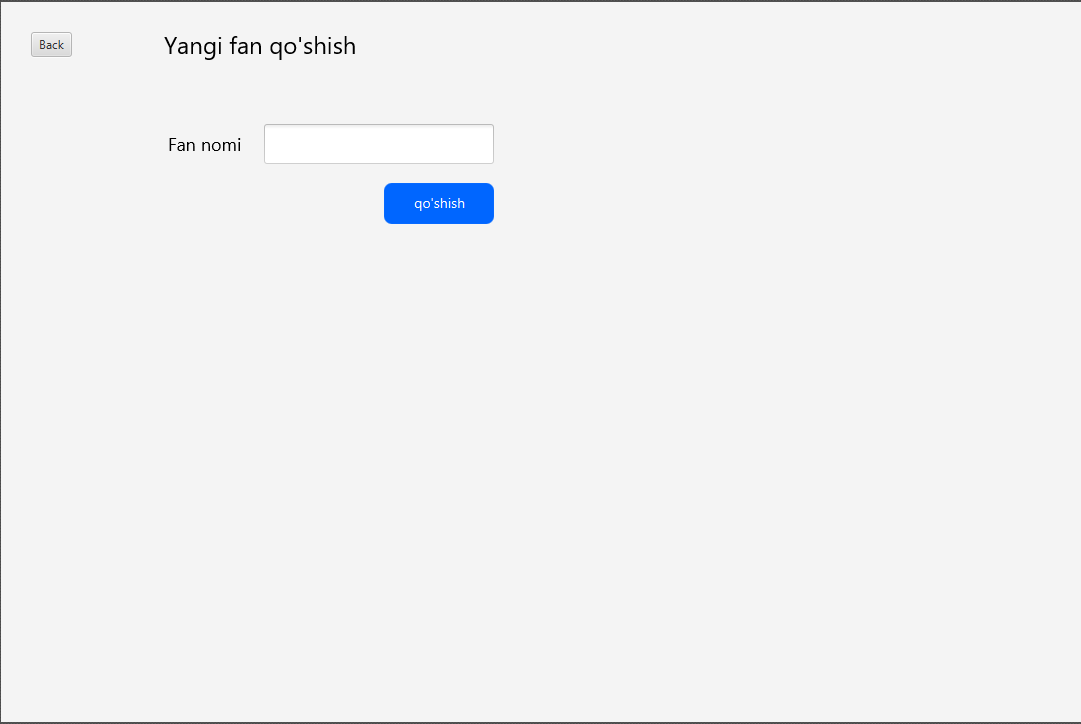
**Fanni o’chirish**

****

**28 - rasm.** Fanni o’chirish oynasi.

Tasdiqlash belgisini belgilaganingizdan so’ngina fan ma’lumotlari o’chiriladi.

**Yangi fan qo’shish**

****

**29 - rasm**. Yangi fan qo’shish oynasi.

Faqatgina fan nomini kiritish kifoya, so’ngra qo’shish tugmasini bossak bas yangi fan qo’shiladi.

**3.5 Foydalaniladigan instrumental vositalar**

|  |  |
| --- | --- |
|  | Intelliji Idea- Java dasturlash tilida daturni ishlab chiqishga yordam beruvchi kod muharririri. |
|  | JavaFX-desktop ilovalar yaratish uchun Java dasturlash tilining kutubxonasi. |
|  | Scene Builder-JavaFX GUI elementlaridan foydalanish uchun va vizual ko’rish uchun insstrumental vositalar jamlanmasi |
|  | StarUML- Ishlab chiqmoqchi bo’lgan dasturimizning diagrammalarini shakllantirishga xizmat qiluvchi dastur. |
|  | MySql- Dastur uchun kerakli ma’lumotlar bazasini shakllantirish va to’g’ri so’rovni shakllantirishga zarur muhit. |
|  | MySql WorkBench- MySqlda ma’lumotlar yozilishi uchun kerakli kod tahrirlovchisi. |

**IV. XULOSA.**

Bu loyiha Java dasturlash tilining JavaFX kutubxonasi yordamida yaratilgan. Bu dastur insonlarga ma’lumotlarni yanada yaxshiroq va samaraliroq eslab qolishga yordam beradi. Dastur faqatgina desktop application bo’lib kompyuter qurilmalarida ishlaydi. Dasturda siz o’zingizga kerakli bo’lgan fanlarni qo’shishingiz va shu fanlarga testlarni biriktirishingiz mumkin bo’ladi.

Testlarni qo’shib bo’lgandan so’ng test jarayonini boshlab ularni yechishingiz mumkin, va o’z natijangizga ham ega bo’lasiz (sonli va foiz ko’rinishda).

Keraksiz fanlarni yoki shunga biriktirilgan testlarni o’chirib tashlash imkoniyatlari ham mavjud. Bundan tashqari ularni yangilash va jadval ko’rinishida ko’rish mumkin.

Dastur ma’lumotlar bazasi bilan to’liq bog’langan. Sizning fanlar yoki testlar ustida qilgan amallaringiz ma’lumotlar bazasida doimiy tarzda saqlanib qoladi.

Dastur ishga tushish jarayonida ma’lumotlar bazasi bilan bog’lanib undagi ma’lumotlarni ekranda chop etadi. Dastur o’zida hech qanday ma’lumotni doimiy saqlab turmaydi. Barchasi ma’lumotlar bazasi orqali bajariladi.

Batafsil ma’lumotlarni dasturning “Dastur haqida” menyusidan ham topishingiz mumkin.

**V. FOYDALANILGAN ADABIYOTLAR.**

1. Kim Topley, “JavaFX Developer’s Guide” // Copyright by Pearson Education, Inc., - 2011. – 1150 p. ISBN 13: 978-0-321-60165-0
2. Mohamed Taman, “JavaFX Essentials” Create amazing Java GUI applications with this hands-on, fast-paced guide // Copyright Packt Publishing, -2015. – 224 p. ISBN 978-1-78439-802-6.
3. Steve Holzner, “Design Patterns for Dummies” // Copyright by Wiley Publishing, Inc., Indianapolis, Indiana, -2006. – 321 p. ISBN-13: 978-0-471- 79854-5
4. Martin Fitzpatrick, “Create Simple GUI Applications, with Python & Qt5”, The hands-on guide to building desktop apps with Python // CC BY-NC-SA © 2015. -264 p.
5. <https://www.geeksforgeeks.org>
6. <https://stackoverflow.com>
7. <https://intranet.ytit.uz>
8. <https://www.javatpoint.com>
9. <https://www.w3schools.com>
10. <https://beginnersbook.com>
11. <https://www.programiz.com>
12. <https://www.tutorialspoint.com>
13. <https://openjfx.io>
14. <https://gluonhq.com/products/javafx>
15. <https://docs.oracle.com/javase/8/javafx/api/toc.htm>
16. <https://www.javatpoint.com/javafx-tutorial>
17. <https://www.jetbrains.com/help/idea/javafx.html>
18. <https://github.com/Rustam-Fozilov/Testbook>

**VI. ILOVALAR**

**Dastur strukturasi**

* **Bo’limlar**
* **Package**
* Controller , fxml , Java classlar , css

|  |  |  |  |
| --- | --- | --- | --- |
| **Dasturdagi Controllerlar** | **Dasturdagi Fxmlslar** | **Java classlar** | **Boshqa** |
| HelloController | Hello-view.fxml | Database | Style.css |
| SubjectsController | subjects.fxml | HelloApplication | **Images** |
| ResultController | result.fxml | QuestionInformation | ... |
| TestStartedController | Test-starter.fxml | SubjectInformation |  |
| SettingsController | settings.fxml |  |  |
| EditController | Edit-test.fxml |  |  |
| AddTestController | Add-test.fxml |  |  |
| AddSubjectController | Add-subject.fxml |  |  |
| DeleteSubjectController | Delete-subject.fxml |  |  |
| InfoController | Info.fxml |  |  |