**KAUNO  TECHNOLOGIJOS  UNIVERSITETAS**

**INFORMATIKOS FAKULTETAS**

**T120B165 Saityno taikomųjų programų projektavimas Tomas Blažauskas**

**Galutinė ataskaita**

Žaidimų svetainė „Gameba“

Atliko:

Lukas Kuzmickas IFF-1/6

Priėmė:

prof. BLAŽAUSKAS Tomas

doc. prakt. Mažutienė Rasa

Turinys

[Sprendžiamo uždavinio aprašymas 3](#_Toc184922006)

[Sistemos paskirtis 3](#_Toc184922007)

[Funkciniai reikalavimai 3](#_Toc184922008)

[Sistemos architektūra (pakanka UML deployment diagramos) 4](#_Toc184922009)

[Naudotojo sąsajos projektas (projektuojamos sąsajos langų wireframe`ai ir juos atitinkančios realizacijos langų iškarpos) 5](#_Toc184922010)

[API specifikacija - struktūriškai aprašyti API naudojant OpenAPI (atskiras api-spec.yaml failas). Taip pat kiekvienam API metodui pateikti: 13](#_Toc184922011)

[Išvados 16](#_Toc184922012)

[Šaltiniai 17](#_Toc184922013)

# Sprendžiamo uždavinio aprašymas

"Gameba" yra žaidimų forumas , forume yra svarbus žaidimai, šie žaidimai turi kategorijas ir komentarus. Taip išreiškiama nuomonė apie kiekvieną kategorijos žaidimą kitiems, prieš kitam jį perkant ir bandant.

## Sistemos paskirtis

Galime lengvai rasti savo norimus žaidimus pagal kategorijas, įvertinti ir išreikšti savo nuomonę apie žaidimus ir leisti kitiems suprasti ką perką ir žaidžią.

## Funkciniai reikalavimai

Sistemos funkciniai reikalavimai :

* Sistema turi tris naudotojo tipus: svečias, naudotojas ir administratorius;
* Visi gali peržiūrėti komentarus arba komentarą;
* Visi gali peržiūrėti kategorijas arba kategoriją;
* Visi gali peržiūrėti žaidimus arba žaidimą;
* Naudotojas arba administratorius gali pridėti kategorijas;
* Administratorius gali redaguoti visas kategorijas;
* Naudotojas gali redaguoti tik savo kategorijas;
* Administratorius gali ištrinti kategorijas ;
* Kategorijos turi žaidimus;
* Žaidimai turi komentarus;
* Naudotojas arba administratorius gali pridėti žaidimus prie kategorijos;
* Naudotojas gali redaguoti tik savo žaidimus ;
* Administratorius gali redaguoti visus žaidimus ;
* Administratorius gali ištrinti žaidimus iš kategorijų ;
* Naudotojas arba administratorius gali komentuoti apie žaidimus;
* Naudotojas gali redaguoti tik savo komentarus ;
* Administratorius gali redaguoti visus komentarus ;
* Administratorius gali ištrinti komentarus ;
* Naudotojai gali prisijungti/atsijungti prie sistemos ;
* Naudotojai gali susikurti paskyrą .

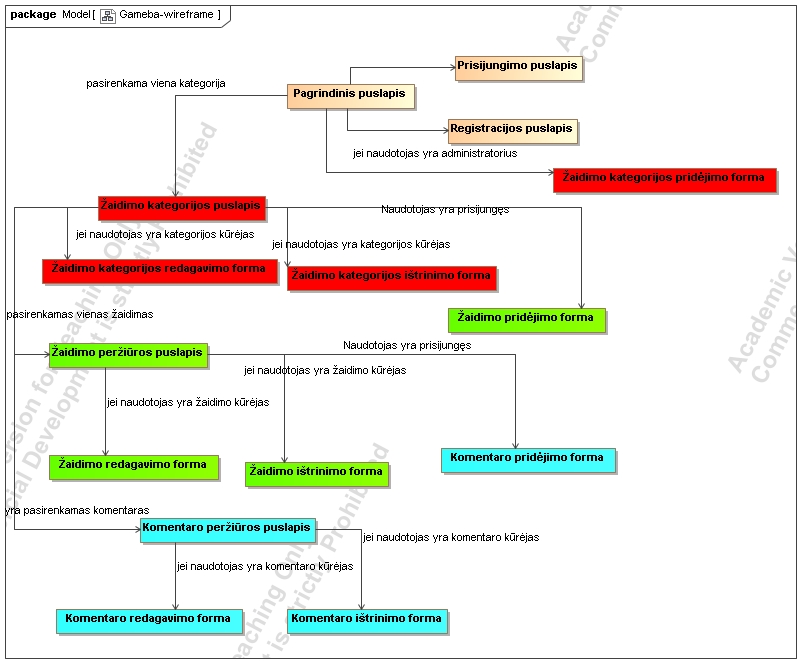
# Sistemos architektūra (pakanka UML deployment diagramos)

*Paveikslėlis, kuriame yra tekstas, ekrano kopija, diagrama

Automatiškai sugeneruotas aprašymas*

*1 pav. Deployment UML diagrama.*

# Naudotojo sąsajos projektas (projektuojamos sąsajos langų wireframe`ai ir juos atitinkančios realizacijos langų iškarpos)



*2 pav. Naudotojo sąsajos projektas.*

Paveikslėlis, kuriame yra tekstas, ekrano kopija, Šriftas, programinė įranga

Automatiškai sugeneruotas aprašymas

*3 pav. Pagrindinis puslapis.*

Paveikslėlis, kuriame yra tekstas, ekrano kopija, programinė įranga, Šriftas

Automatiškai sugeneruotas aprašymas

*4 pav. Prisijungimo puslapis.*

Paveikslėlis, kuriame yra tekstas, ekrano kopija, programinė įranga, Tinklalapis

Automatiškai sugeneruotas aprašymas

*5 pav. Registracijos puslapis.*

*Paveikslėlis, kuriame yra tekstas, ekrano kopija, Šriftas, Tinklalapis

Automatiškai sugeneruotas aprašymas*

*6 pav. Kategorijos peržiūros puslapis.*

*Paveikslėlis, kuriame yra tekstas, ekrano kopija, Šriftas

Automatiškai sugeneruotas aprašymas*

*7 pav. Kategorijos ištrinimo forma.*

*Paveikslėlis, kuriame yra tekstas, ekrano kopija, Šriftas

Automatiškai sugeneruotas aprašymas*

*8 pav. Kategorijos redagavimo forma.*

*Paveikslėlis, kuriame yra tekstas, ekrano kopija, Šriftas, skaičius

Automatiškai sugeneruotas aprašymas*

*9 pav. Žaidimo pridėjimo forma.*

*Paveikslėlis, kuriame yra tekstas, ekrano kopija, programinė įranga, Kompiuterio piktograma

Automatiškai sugeneruotas aprašymas*

*10 pav. Žaidimo peržiūros puslapis.*

*Paveikslėlis, kuriame yra tekstas, ekrano kopija, Šriftas, programinė įranga

Automatiškai sugeneruotas aprašymas*

*11 pav. Žaidimo ištrinimo forma.*

*Paveikslėlis, kuriame yra tekstas, ekrano kopija, Šriftas, programinė įranga

Automatiškai sugeneruotas aprašymas*

*12 pav. Žaidimo ištrinimo forma.*

*Paveikslėlis, kuriame yra tekstas, ekrano kopija, linija, Šriftas

Automatiškai sugeneruotas aprašymas*

*13 pav. Žaidimo redagavimo forma.*

*Paveikslėlis, kuriame yra tekstas, ekrano kopija, Šriftas, linija

Automatiškai sugeneruotas aprašymas*

*14 pav. Komentaro pridėjimas.*

*Paveikslėlis, kuriame yra tekstas, vizitinė kortelė, ekrano kopija, dizainas

Automatiškai sugeneruotas aprašymas*

*15 pav. Komentaro peržiūros puslapis.*

*Paveikslėlis, kuriame yra tekstas, ekrano kopija, Šriftas

Automatiškai sugeneruotas aprašymas*

*16 pav. Komentaro ištrinimo forma.*

*Paveikslėlis, kuriame yra tekstas, ekrano kopija, Šriftas

Automatiškai sugeneruotas aprašymas*

*17 pav. Komentaro redagavimo forma.*

# API specifikacija - struktūriškai aprašyti API naudojant OpenAPI (atskiras api-spec.yaml failas). Taip pat kiekvienam API metodui pateikti:

**Gameba API Įvadas**

**Gameba API** suteikia galimybę tvarkyti žaidimus, kategorijas ir komentarus naudojant RESTful sąsają. API palaiko užklausas išteklių gavimui, kūrimui, atnaujinimui ir ištrynimui. Žemiau pateikiama pagrindinių galimų endpoint'ų ir jų atsakymų kodų apžvalga:

**Kategorijos**

* **GET /categories**: Gauti visų kategorijų sąrašą.  
  **Atsakymo kodai**:
  + 200: Sėkmingas gavimas.
* **POST /categories**: Sukurti naują kategoriją.  
  **Atsakymo kodai**:
  + 201: Kategorija sukurta.
  + 400: Klaidinga užklausa.
  + 422: Nepavyksta apdoroti užklausos.
* **GET /categories/{categoryId}**: Gauti kategoriją pagal ID.  
  **Atsakymo kodai**:
  + 200: Kategorija rasta.
  + 404: Kategorija nerasta.
* **PUT /categories/{categoryId}**: Atnaujinti kategoriją pagal ID.  
  **Atsakymo kodai**:
  + 200: Kategorija atnaujinta.
  + 400: Klaidinga užklausa.
  + 404: Kategorija nerasta.
  + 422: Nepavyksta apdoroti užklausos.
* **DELETE /categories/{categoryId}**: Ištrinti kategoriją pagal ID.  
  **Atsakymo kodai**:
  + 204: Kategorija ištrinta.
  + 404: Kategorija nerasta.

**Žaidimai**

* **GET /categories/{categoryId}/games**: Gauti žaidimus, priklausančius konkrečiai kategorijai.  
  **Atsakymo kodai**:
  + 200: Sėkmingas gavimas.
* **POST /categories/{categoryId}/games**: Pridėti naują žaidimą prie kategorijos.  
  **Atsakymo kodai**:
  + 201: Žaidimas sukurtas.
  + 400: Klaidinga užklausa.
  + 422: Nepavyksta apdoroti užklausos.
* **GET /categories/{categoryId}/games/{gameId}**: Gauti žaidimą pagal ID konkrečioje kategorijoje.  
  **Atsakymo kodai**:
  + 200: Žaidimas rastas.
  + 404: Žaidimas nerastas.
* **PUT /categories/{categoryId}/games/{gameId}**: Atnaujinti žaidimą pagal ID.  
  **Atsakymo kodai**:
  + 200: Žaidimas atnaujintas.
  + 400: Klaidinga užklausa.
  + 404: Žaidimas nerastas.
  + 422: Nepavyksta apdoroti užklausos.
* **DELETE /categories/{categoryId}/games/{gameId}**: Ištrinti žaidimą pagal ID.  
  **Atsakymo kodai**:
  + 204: Žaidimas ištrintas.
  + 404: Žaidimas nerastas.

**Komentarai**

* **GET /categories/{categoryId}/games/{gameId}/comments**: Gauti komentarus apie konkretų žaidimą.  
  **Atsakymo kodai**:
  + 200: Sėkmingas gavimas.
* **POST /categories/{categoryId}/games/{gameId}/comments**: Pridėti komentarą prie žaidimo.  
  **Atsakymo kodai**:
  + 201: Komentaras sukurtas.
  + 400: Klaidinga užklausa.
  + 422: Nepavyksta apdoroti užklausos.

openapi: 3.0.0

info:

  title: Gameba API

  description: API for managing games and categories.

  version: 1.0.0

servers:

  - url: https://squid-app-xuzxl.ondigitalocean.app/api/v1

paths:

  /categories:

    get:

      summary: Get all categories

      operationId: getAllCategories

      responses:

        "200":

          description: List of categories

          content:

            application/json:

              schema:

                type: array

                items:

                  $ref: "#/components/schemas/Category"

    post:

      summary: Create a new category

      operationId: createCategory

      requestBody:

        required: true

        content:

          application/json:

            schema:

              $ref: "#/components/schemas/CreateCategory"

      responses:

        "201":

          description: Category created

          content:

            application/json:

              schema:

                $ref: "#/components/schemas/Category"

        "400":

          description: Bad request

  /categories/{categoryId}:

    get:

      summary: Get a category by ID

      operationId: getCategory

      parameters:

        - $ref: '#/components/parameters/categoryId'

      responses:

        "200":

          description: A category

          content:

            application/json:

              schema:

                $ref: "#/components/schemas/Category"

        "404":

          description: Category not found

    put:

      summary: Update a category by ID

      operationId: updateCategory

      parameters:

        - $ref: '#/components/parameters/categoryId'

      requestBody:

        required: true

        content:

          application/json:

            schema:

              $ref: "#/components/schemas/UpdateCategory"

      responses:

        "200":

          description: Updated category

        "404":

          description: Category not found

    delete:

      summary: Delete a category by ID

      operationId: deleteCategory

      parameters:

        - $ref: '#/components/parameters/categoryId'

      responses:

        "204":

          description: Category deleted

  /categories/{categoryId}/games:

    get:

      summary: Get all games in a category

      operationId: getGamesByCategory

      parameters:

        - $ref: '#/components/parameters/categoryId'

      responses:

        "200":

          description: List of games

          content:

            application/json:

              schema:

                type: array

                items:

                  $ref: "#/components/schemas/Game"

    post:

      summary: Create a new game in a category

      operationId: createGame

      parameters:

        - $ref: '#/components/parameters/categoryId'

      requestBody:

        required: true

        content:

          application/json:

            schema:

              $ref: "#/components/schemas/CreateGame"

      responses:

        "201":

          description: Game created

          content:

            application/json:

              schema:

                $ref: "#/components/schemas/Game"

        "400":

          description: Bad request

  /categories/{categoryId}/games/{gameId}:

    get:

      summary: Get a game by ID

      operationId: getGame

      parameters:

        - $ref: '#/components/parameters/categoryId'

        - $ref: '#/components/parameters/gameId'

      responses:

        "200":

          description: A game

          content:

            application/json:

              schema:

                $ref: "#/components/schemas/Game"

        "404":

          description: Game not found

    put:

      summary: Update a game by ID

      operationId: updateGame

      parameters:

        - $ref: '#/components/parameters/categoryId'

        - $ref: '#/components/parameters/gameId'

      requestBody:

        required: true

        content:

          application/json:

            schema:

              $ref: "#/components/schemas/UpdateGame"

      responses:

        "200":

          description: Updated game

    delete:

      summary: Delete a game by ID

      operationId: deleteGame

      parameters:

        - $ref: '#/components/parameters/categoryId'

        - $ref: '#/components/parameters/gameId'

      responses:

        "204":

          description: Game deleted

  /categories/{categoryId}/games/{gameId}/comments:

    get:

      summary: Get comments for a game

      operationId: getCommentsForGame

      parameters:

        - $ref: '#/components/parameters/categoryId'

        - $ref: '#/components/parameters/gameId'

      responses:

        "200":

          description: List of comments

          content:

            application/json:

              schema:

                type: array

                items:

                  $ref: "#/components/schemas/Comment"

    post:

      summary: Add a comment for a game

      operationId: createComment

      parameters:

        - $ref: '#/components/parameters/categoryId'

        - $ref: '#/components/parameters/gameId'

      requestBody:

        required: true

        content:

          application/json:

            schema:

              $ref: "#/components/schemas/CreateComment"

      responses:

        "201":

          description: Comment created

        "400":

          description: Bad request

components:

  schemas:

    Category:

      type: object

      properties:

        id:

          type: integer

        name:

          type: string

    CreateCategory:

      type: object

      required:

        - name

      properties:

        name:

          type: string

    UpdateCategory:

      type: object

      properties:

        name:

          type: string

    Game:

      type: object

      properties:

        id:

          type: integer

        title:

          type: string

        description:

          type: string

    CreateGame:

      type: object

      required:

        - title

        - description

      properties:

        title:

          type: string

        description:

          type: string

    UpdateGame:

      type: object

      properties:

        title:

          type: string

        description:

          type: string

    Comment:

      type: object

      properties:

        id:

          type: integer

        content:

          type: string

    CreateComment:

      type: object

      required:

        - content

      properties:

        content:

          type: string

  parameters:

    categoryId:

      name: categoryId

      in: path

      required: true

      schema:

        type: integer

    gameId:

      name: gameId

      in: path

      required: true

      schema:

        type: integer

  responses:

    badRequest:

      description: Bad request

# Išvados

Apibendrinant, žaidimų svetainė „Gameba“ buvo sukurta naudojant Saityno taikomųjų programų projektavimo principus, mūsų svetainei turime atskirą API serverį ir Frontend pusę, kurioje naudojame šitą serverį. API integracija tampa labai lengva, nereikia per daug kodo Frontend dalyje ir viską galime atlikti daug lengviau ir greičiau. Tai pagreitina projekto kūrimo spartą.

# Šaltiniai

1. Projekto Git saugykla: [Gameba](https://github.com/lukkuz1/Gameba)
2. Gyvas padeployintas projektas (front-end): [Gameba-fe](https://squid-app-2-crr64.ondigitalocean.app/)
3. Padeployintas backend (API): [Gameba-be](https://squid-app-xuzxl.ondigitalocean.app/)