

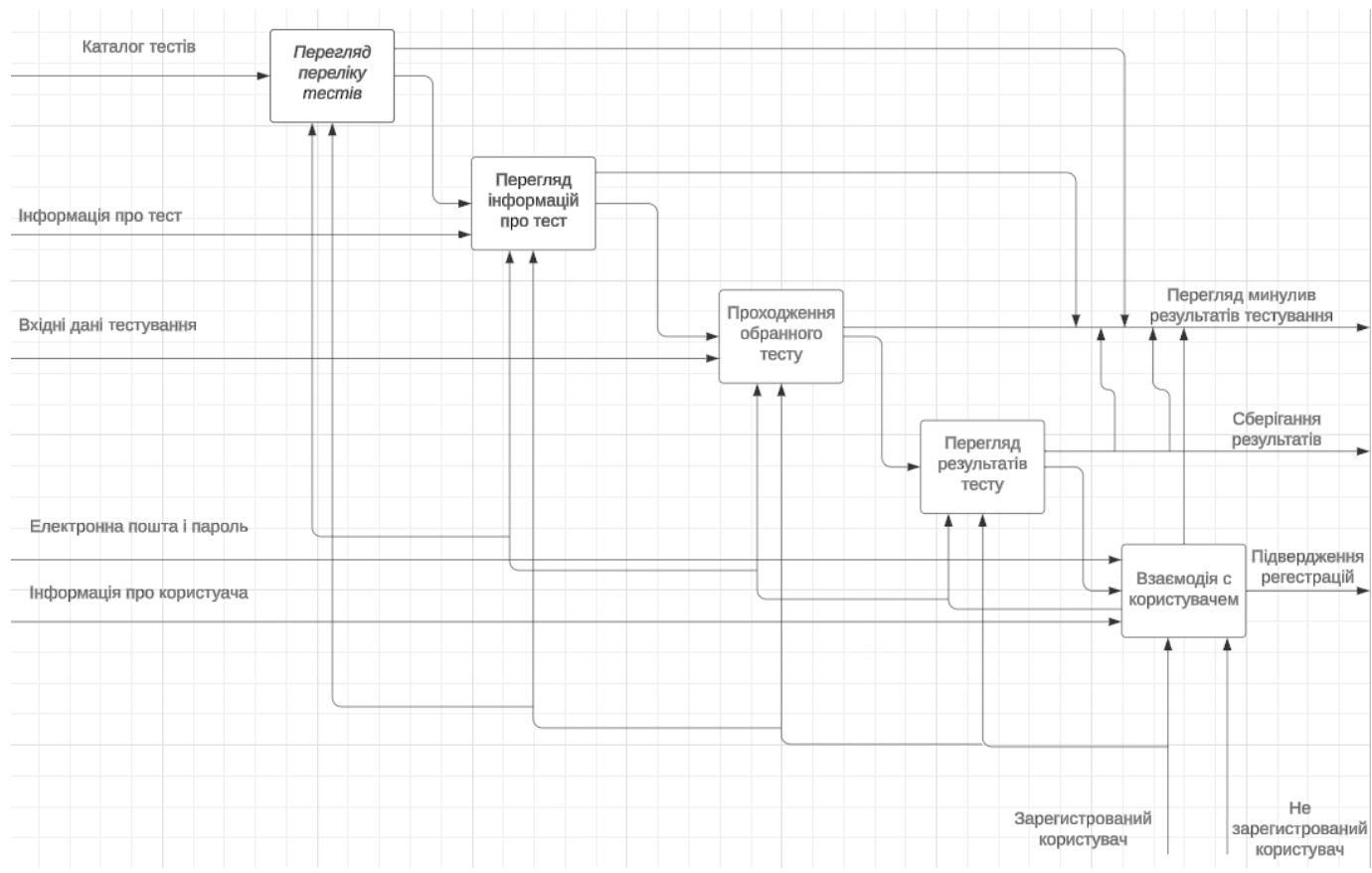
## Додаток А

### ГРАФІЧНІ МАТЕРІАЛИ

## КОНЦЕПТУАЛЬНА ДІАГРАМА



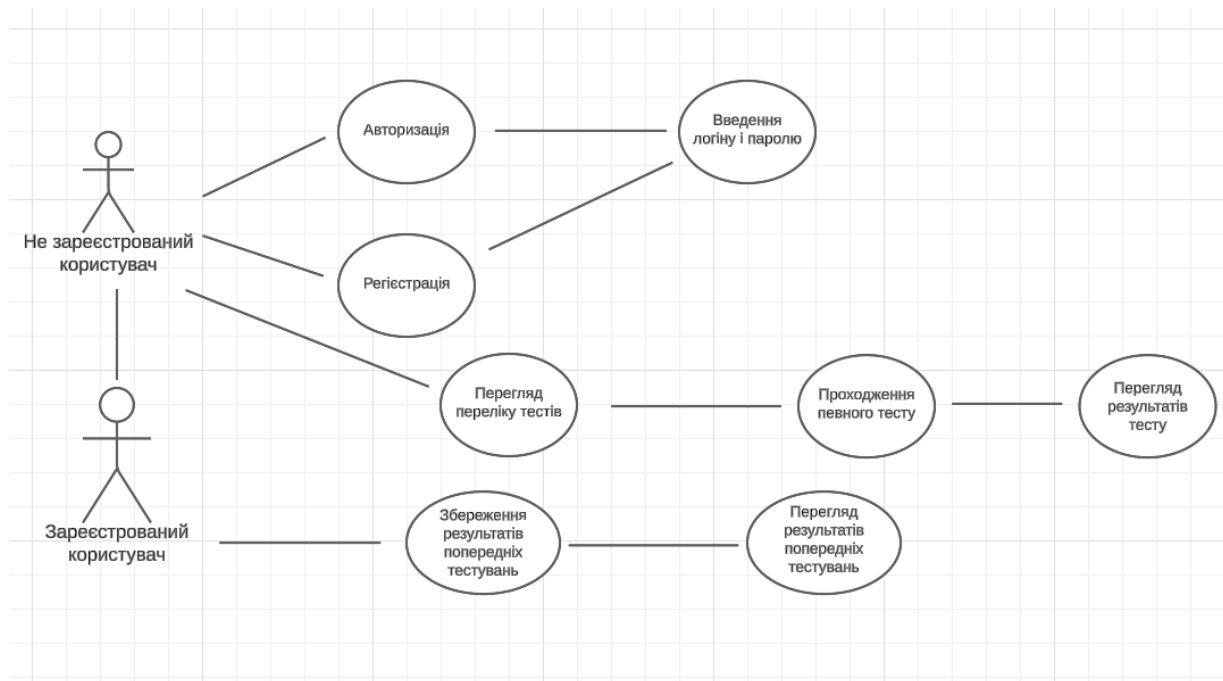
## ДЕКОМПОЗИЦІЯ КОНЦЕПТУАЛЬНОЇ ДІАГРАМИ



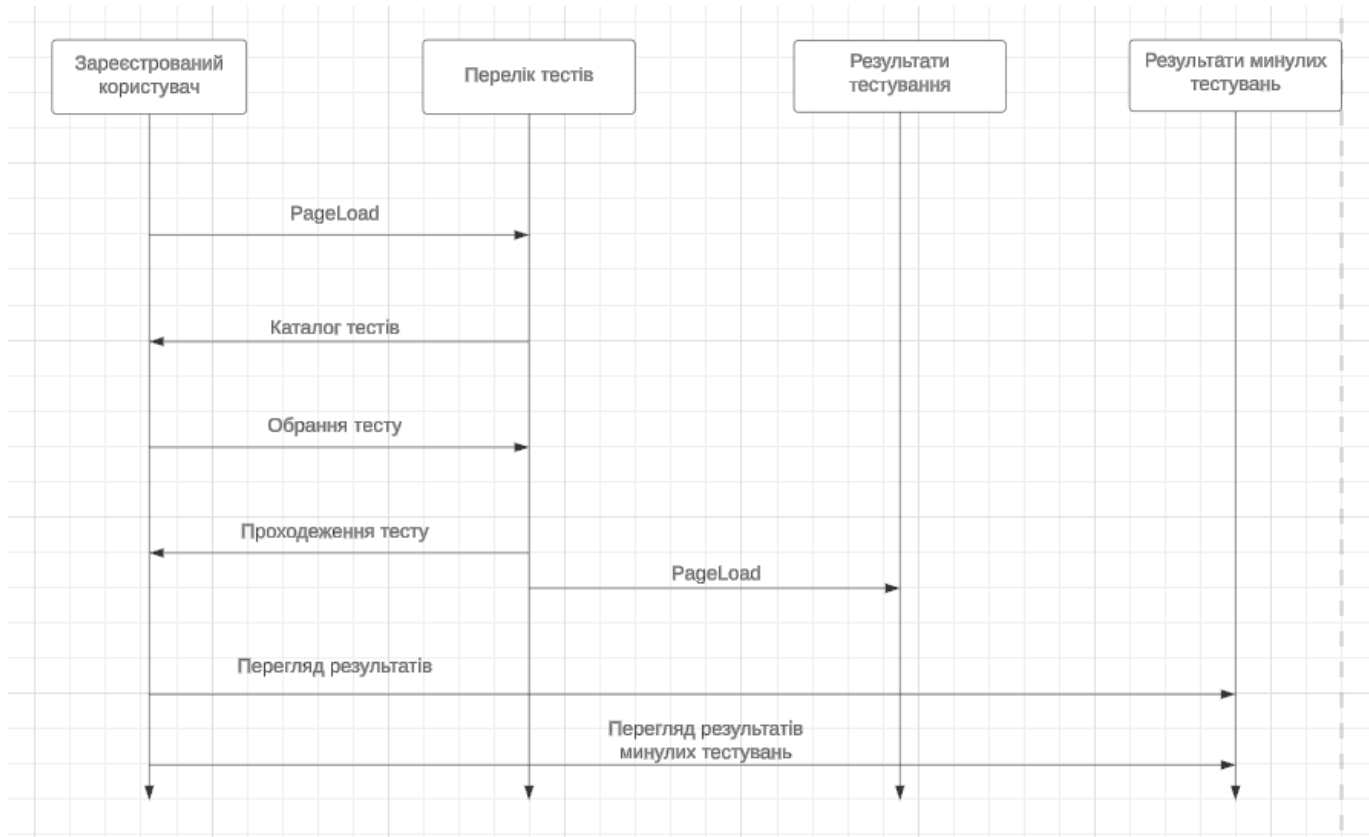
## КОНТЕКСТНА ДІАГРАМА DFD



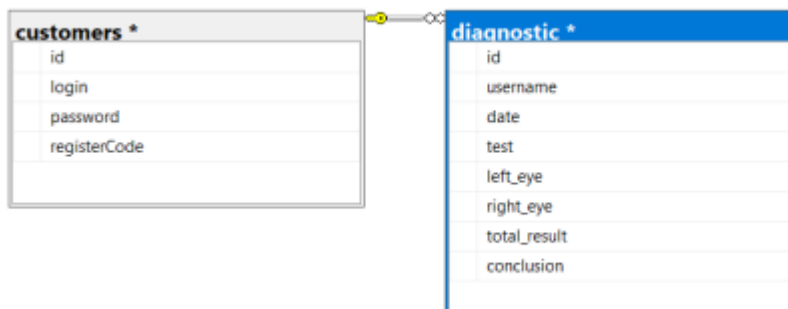
## ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ СИСТЕМИ



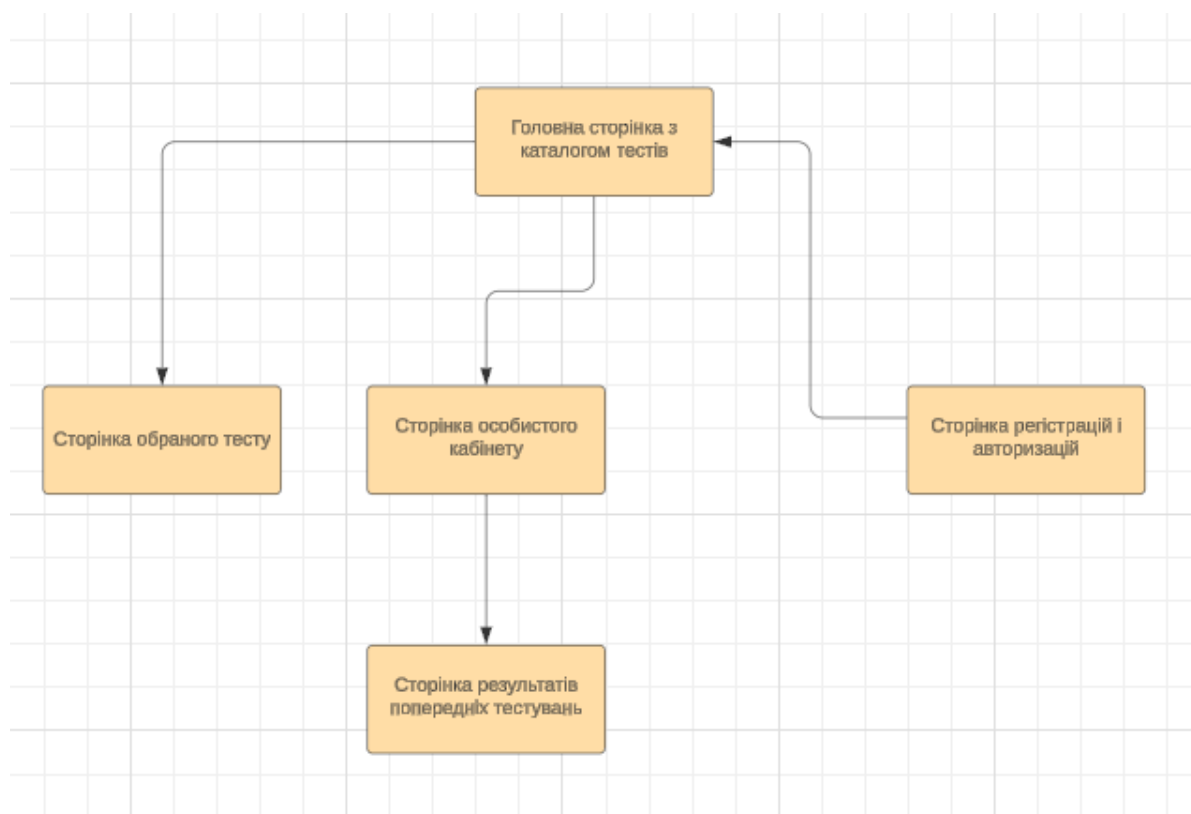
# ДІАГРАМА ПОСЛІДОВНОСТІ ДІЙ СИСТЕМИ



## ДІАГРАМА БАЗИ ДАНИХ



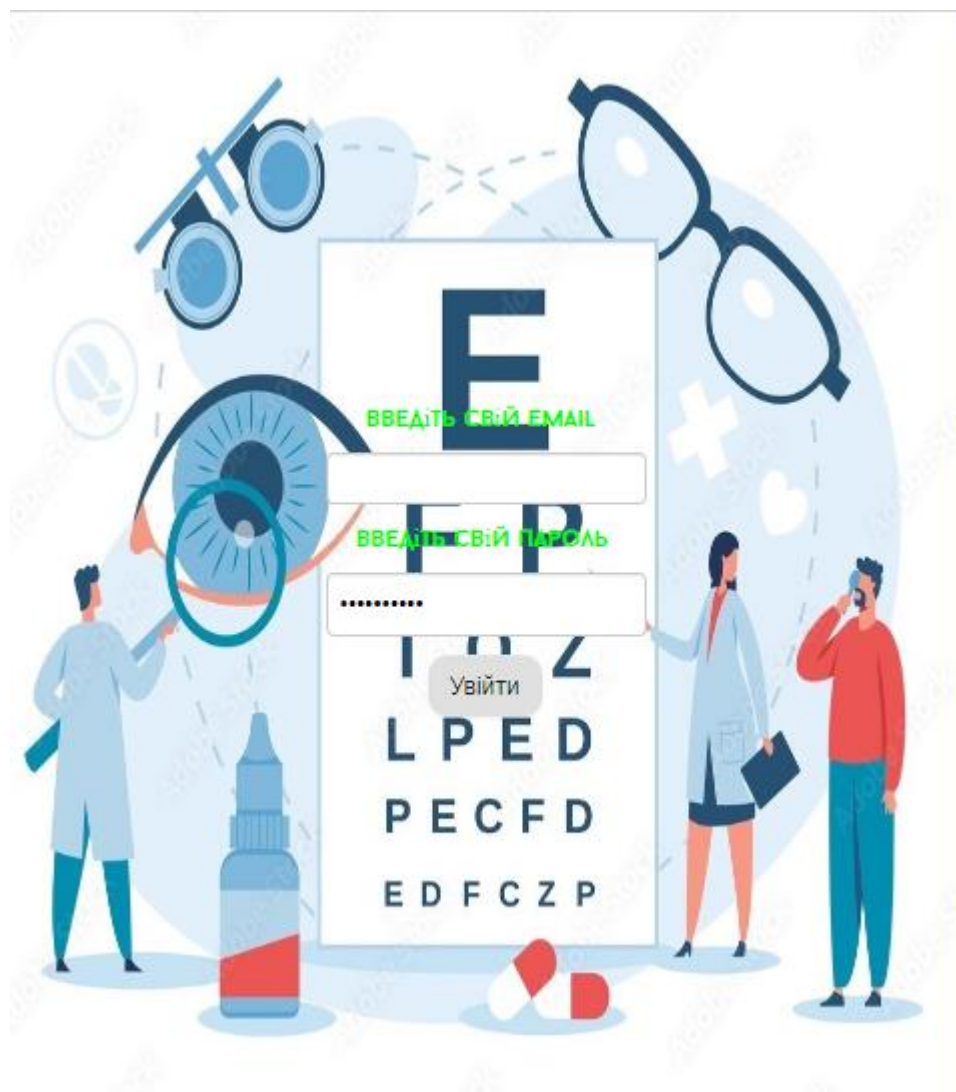
## КАРТА ДОДАТКУ



## СТОРІНКА РЕЄСТРАЦІЇ



## СТОРІНКА АВТОРИЗАЦІЇ



## ГОЛОВНА СТОРІНКА ДОДАТКУ

Привіт,  
123@gmail.com

Особистий кабінет

Тест на перевірку гостроти зору за допомогою таблиці Сивцева

Тест на перевірку кольоровідчуття за допомогою таблиці Р'ябкіна

Перейти до тесту

Перейти до тесту

Тест на перевірку контрастної чутливості за допомогою таблиці Пеллі-Робсон

Тест на діагностику рефракції ока за допомогою дуохромного тесту

## СТОРІНКА ОБРАНОГО ТЕСТУ

Привіт,  
123@gmail.com

Особистий кабінет

Тест на перевірку гостроти зору за допомогою таблиці Сивцева.  
Рекомендація перед проходженням тесту:  
1. Тримати екран телефону на відстані витягнутої руки щоб результати тестування були правдивими. 2. Освітлення в приміщенні повинно відповідати нормативним нормам. 3. Для правдивих результатів рекомендується пройти тестування 2 рази.

Ш Б  
М Н К  
Ы М Б Ш  
Б Ы Н К М  
И Н Ш М К  
К М И Н Б  
.....  
.....  
.....

Почати тестування



## СТОРІНКА ПРОХОДЖЕННЯ ОБРАНОГО ТЕСТУ



## СТОРІНКА ОСОБИСТОГО КАБІНЕТУ

Особистий кабінет

123@gmail.com

### Список минулих тестувань

14.06.2024 16:41: Тест Сивцева

[Переглянути результати](#)

14.06.2024 17:05: Тест Дуохромний

[Переглянути результати](#)



## Додаток Б

### КЕРІВНИЦТВО КОРИСТУВАЧА

## ВСТУП

Даний документ являє собою робочу документацію додатку для контролю та тестуванню якості розу очей. Робоча документація включає: Керівництво користувача для використання додатку для контролю та тестуванню якості розу очей. Користувач додатку може переглянути каталог тестів. Далі користувач може обрати вибраний тест. Після цього він зможе почати проходити тестування у вибраному тесті. Потім він може подивитися результати свого тестування. Також через головну сторінку він може перейти в особистий кабінет (якщо він авторизован) і подивитися результати своїх попередніх тестувань.

## A.1 ПРОГРАМНІ ТА АПАРАТНІ ВИМОГИ ДО ДОДАТКА

Ціллю розробки мобільного додатку є надання користувачам можливості проходити тестування стану зору і контролю за цим.

Виходячи з цієї мети, можна сформулювати наступні вимоги до серверної частини:

- операційна система: Windows 7 і вище;
- React та React Native: React версії 12.x або новіше та React Native версії 6.x або новіше;
- редактор коду: Visual Studio Code;
- система контролю версій: Git для контролю версій коду.

До клієнтської частини вимоги є наступні:

- пристрій: мобільний телефон с встановленим додатком і камерою в наявності.
- інтернет-з'єднання: стабільне підключення до інтернету для завантаження даних о користувачі.

## А.2 ПІДГОТОВКА ДО РОБОТИ

Встановлення додатку відбувається через арк-файл який був попередньо завантажений на пристрій. Для запуску, треба натиснути на іконку додатку який з'явиться після встановлення. Після цього ви потратите на сторінку входу у додаток. (рисунок А.2.1).

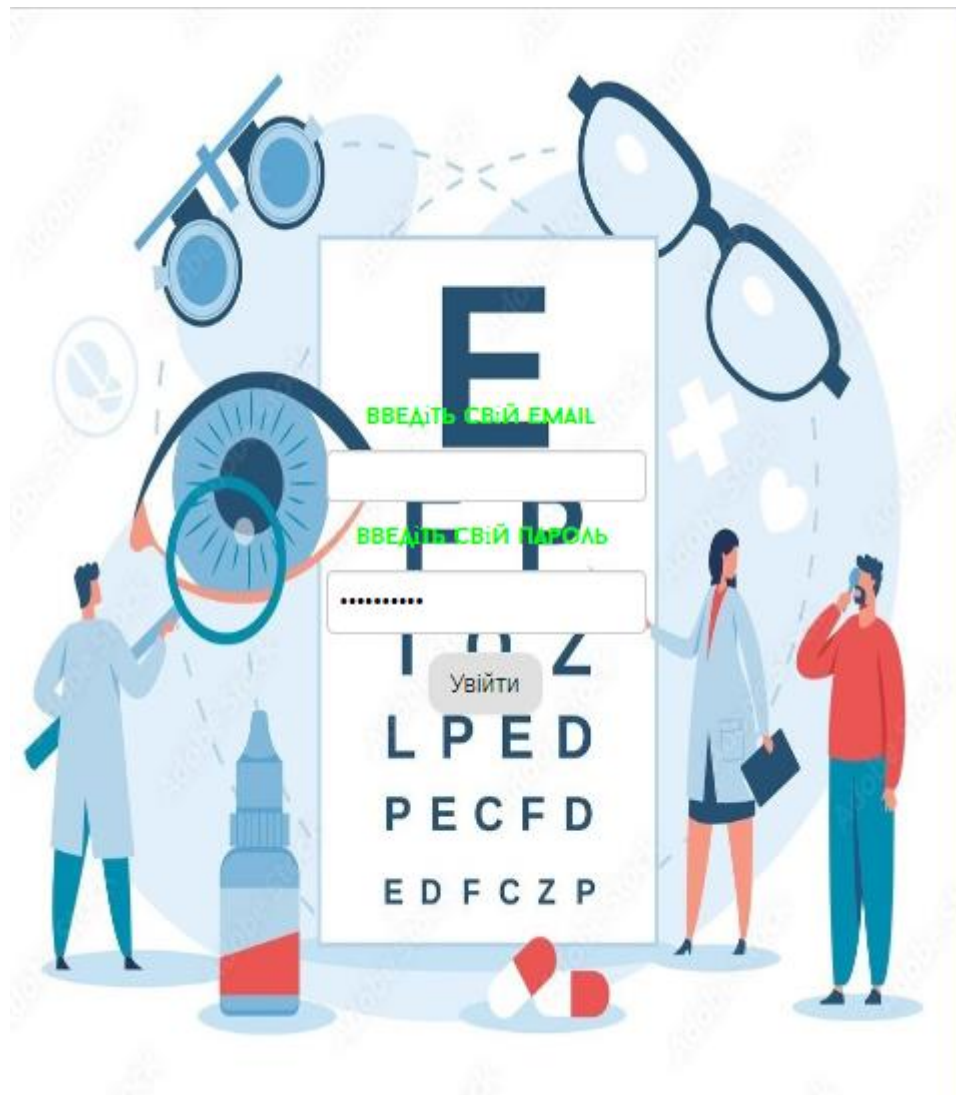


Рисунок А.2.1 – сторінка входу у додаток

### А.3 РЕКОМЕНДАЦІЇ З ОСВОЄННЯ

При запуску додатку користувач потрапляє на сторінку входу у додаток де може пройти авторизацію якщо він вже зареєстрован, або зареєструватися якщо ще цього не зробив. (рисунок А.3.1).



Рисунок А.3.1 – сторінка реєстрацій у додаток

Далі користувач потрапляє на головну сторінку додатку де може обрати тест для проходження із запропонованого переліку або перейти в особистий кабінет щоб подивитися результати своїх минулих тестувань. (рисунк А.3.2)



Рисунок А.3.2 – головна сторінка додатку

Далі якщо користувач обирає тест для проходження на екрані з'являється інформація про обраний тест та кнопка для початку проходження тесту. (рисунок А.3.3)



Рисунок А.3.3 – сторінка інформацій про обраний тест

Далі іде сам процес проходження тесту. (рисунок А.3.4)



Рисунок А.3.4 – сторінка проходження тесту

Якщо користувач переходить в особистий кабінет то там він може продивитися результати своїх минулих тестувань. (рисунок А.3.5)

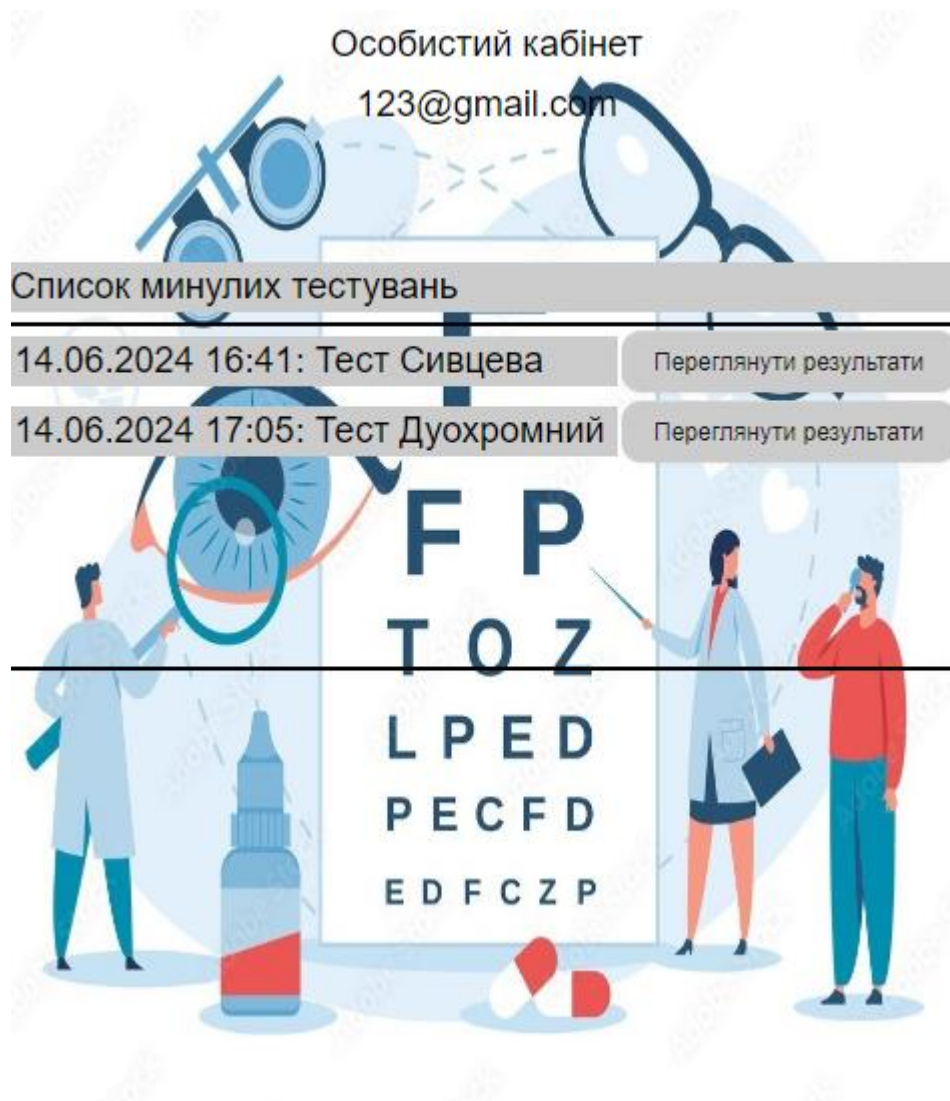


Рисунок А.3.5 – сторінка особистого кабінету

## А.4 АВАРІЙНІ СИТУАЦІЇ

Після виникнення проблем з функціонуванням апаратного забезпечення, нормалізація роботи додатку для контролю та тестуванню якості розу очей відбувається за такою послідовністю

- Перезавантаження пристрою (ОС);
- Запуск додатка.

У випадку помилок з апаратною частиною, крім носіїв даних та програм, відновлення функціональності додатку залежить від ОС. Якщо виникають помилки, пов'язані з програмним забезпеченням, вони також виправляються через ОС. При неправильних діях користувачів, некоректних форматах або недопустимих значеннях введених даних, система виводить відповідні повідомлення для користувача та повертається до попереднього робочого стану після невірних команд або некоректного введення даних. Для успішного використання інформаційної системи рекомендується користуватися робочою документацією. Ця документація призначена для досвідчених користувачів мобільних пристроїв.

## Додаток В

ТЕКСТ ПРОГРАМИ

```
"use strict";

const fs = require("fs");
const path = require("path");
const webpack = require("webpack");
const resolve = require("resolve");
const PnpWebpackPlugin = require("pnp-webpack-plugin");
const HtmlWebpackPlugin = require("html-webpack-plugin");
const CaseSensitivePathsPlugin = require("case-sensitive-paths-webpack-plugin");
const InlineChunkHtmlPlugin = require("react-dev-utils/InlineChunkHtmlPlugin");
const TerserPlugin = require("terser-webpack-plugin");
const MiniCssExtractPlugin = require("mini-css-extract-plugin");
const OptimizeCSSAssetsPlugin = require("optimize-css-assets-webpack-plugin");
const safePostCssParser = require("postcss-safe-parser");
const ManifestPlugin = require("webpack-manifest-plugin");
const InterpolateHtmlPlugin = require("react-dev-utils/InterpolateHtmlPlugin");
const WorkboxWebpackPlugin = require("workbox-webpack-plugin");
const WatchMissingNodeModulesPlugin = require("react-dev-
utils/WatchMissingNodeModulesPlugin");
const ModuleScopePlugin = require("react-dev-utils/ModuleScopePlugin");
const getCSSModuleLocalIdent = require("react-dev-utils/getCSSModuleLocalIdent");
const paths = require("./paths");
const getClientEnvironment = require("./env");
const ModuleNotFoundPlugin = require("react-dev-utils/ModuleNotFoundPlugin");
const ForkTsCheckerWebpackPlugin = require("react-dev-
utils/ForkTsCheckerWebpackPlugin");
const typescriptFormatter = require("react-dev-utils/typescriptFormatter");
```

```

// Source maps are resource heavy and can cause out of memory issue for large source
files.
const shouldUseSourceMap = process.env.GENERATE_SOURCEMAP !== "false";
// Some apps do not need the benefits of saving a web request, so not inlining the chunk
// makes for a smoother build process.
const shouldInlineRuntimeChunk = process.env.INLINE_RUNTIME_CHUNK !==
"false";

// Check if TypeScript is setup
const useTypeScript = fs.existsSync(paths.appTsConfig);

// style files regexes
const cssRegex = /\.css$/;
const cssModuleRegex = /\.module\.css$/;
const sassRegex = /\.(scss|sass)$/;
const sassModuleRegex = /\.module\.(scss|sass)$/;

// This is the production and development configuration.
// It is focused on developer experience, fast rebuilds, and a minimal bundle.
module.exports = function(webpackEnv) {
  const isEnvDevelopment = webpackEnv === "development";
  const isEnvProduction = webpackEnv === "production";

  // Webpack uses `publicPath` to determine where the app is being served from.
  // It requires a trailing slash, or the file assets will get an incorrect path.
  // In development, we always serve from the root. This makes config easier.
  const publicPath = isEnvProduction
    ? paths.servedPath

```

```

: isEnvDevelopment && "/";
// Some apps do not use client-side routing with pushState.
// For these, "homepage" can be set to "." to enable relative asset paths.
const shouldUseRelativeAssetPaths = publicPath === "./";

// `publicUrl` is just like `publicPath`, but we will provide it to our app
// as %PUBLIC_URL% in `index.html` and `process.env.PUBLIC_URL` in JavaScript.
// Omit trailing slash as %PUBLIC_URL%/xyz looks better than %PUBLIC_URL%xyz.
const publicUrl = isEnvProduction
  ? publicPath.slice(0, -1)
  : isEnvDevelopment && "";
// Get environment variables to inject into our app.
const env = getClientEnvironment(publicUrl);

// common function to get style loaders
const getStyleLoaders = (cssOptions, preProcessor) => {
  const loaders = [
    isEnvDevelopment && require.resolve("style-loader"),
    isEnvProduction && {
      loader: MiniCssExtractPlugin.loader,
      options: Object.assign(
        {},
        shouldUseRelativeAssetPaths ? { publicPath: "../.." } : undefined
      )
    },
    {
      loader: require.resolve("css-loader"),
      options: cssOptions
    }
  ]

```

```

},
{
  // Options for PostCSS as we reference these options twice
  // Adds vendor prefixing based on your specified browser support in
  // package.json
  loader: require.resolve("postcss-loader"),
  options: {
    // Necessary for external CSS imports to work
    // https://github.com/facebook/create-react-app/issues/2677
    ident: "postcss",
    plugins: () => [
      require("postcss-flexbugs-fixes"),
      require("postcss-preset-env")({
        autoprefixer: {
          flexbox: "no-2009"
        },
        stage: 3
      })
    ],
    sourceMap: isEnvProduction && shouldUseSourceMap
  }
}
].filter(Boolean);
if (preProcessor) {
  loaders.push({
    loader: require.resolve(preProcessor),
    options: {
      sourceMap: isEnvProduction && shouldUseSourceMap
    }
  });
}

```

```

    }
  });
}
return loaders;
};

return {
  mode: isEnvProduction ? "production" : isEnvDevelopment && "development",
  // Stop compilation early in production
  bail: isEnvProduction,
  devtool: isEnvProduction
    ? shouldUseSourceMap
      ? "source-map"
      : false
    : isEnvDevelopment && "cheap-module-source-map",
  // These are the "entry points" to our application.
  // This means they will be the "root" imports that are included in JS bundle.
  entry: [
    // Include an alternative client for WebpackDevServer. A client's job is to
    // connect to WebpackDevServer by a socket and get notified about changes.
    // When you save a file, the client will either apply hot updates (in case
    // of CSS changes), or refresh the page (in case of JS changes). When you
    // make a syntax error, this client will display a syntax error overlay.
    // Note: instead of the default WebpackDevServer client, we use a custom one
    // to bring better experience for Create React App users. You can replace
    // the line below with these two lines if you prefer the stock client:
    // require.resolve('webpack-dev-server/client') + '?!',
    // require.resolve('webpack/hot/dev-server'),

```

```

isEnvDevelopment &&
  require.resolve("react-dev-utils/webpackHotDevClient"),
// Finally, this is your app's code:
paths.appIndexJs
// We include the app code last so that if there is a runtime error during
// initialization, it doesn't blow up the WebpackDevServer client, and
// changing JS code would still trigger a refresh.
].filter(Boolean),
output: {
  // The build folder.
  path: isEnvProduction ? paths.appBuild : undefined,
  // Add /* filename */ comments to generated require()s in the output.
  pathinfo: isEnvDevelopment,
  // There will be one main bundle, and one file per asynchronous chunk.
  // In development, it does not produce real files.
  filename: isEnvProduction
    ? "static/js/[name].[contenthash:8].js"
    : isEnvDevelopment && "static/js/bundle.js",
  // There are also additional JS chunk files if you use code splitting.
  chunkFilename: isEnvProduction
    ? "static/js/[name].[contenthash:8].chunk.js"
    : isEnvDevelopment && "static/js/[name].chunk.js",
  // We inferred the "public path" (such as / or /my-project) from homepage.
  // We use "/" in development.
  publicPath: publicPath,
  // Point sourcemap entries to original disk location (format as URL on Windows)
  devtoolModuleFilenameTemplate: isEnvProduction
    ? info =>

```

```

    path
      .relative(paths.appSrc, info.absoluteResourcePath)
      .replace(/\\/g, "/")
: isEnvDevelopment &&
  (info => path.resolve(info.absoluteResourcePath).replace(/\\/g, "/"))
},
optimization: {
  minimize: isEnvProduction,
  minimizer: [
    // This is only used in production mode
    new TerserPlugin({
      terserOptions: {
        parse: {
          // we want terser to parse ecma 8 code. However, we don't want it
          // to apply any minification steps that turns valid ecma 5 code
          // into invalid ecma 5 code. This is why the 'compress' and 'output'
          // sections only apply transformations that are ecma 5 safe
          // https://github.com/facebook/create-react-app/pull/4234
          ecma: 8
        },
        compress: {
          ecma: 5,
          warnings: false,
          // Disabled because of an issue with Uglify breaking seemingly valid code:
          // https://github.com/facebook/create-react-app/issues/2376
          // Pending further investigation:
          // https://github.com/mishoo/UglifyJS2/issues/2011
          comparisons: false,

```

```
// Disabled because of an issue with Terser breaking valid code:
// https://github.com/facebook/create-react-app/issues/5250
// Pending further investigation:
// https://github.com/terser-js/terser/issues/120
  inline: 2
},
mangle: {
  safari10: true
},
output: {
  ecma: 5,
  comments: false,
  // Turned on because emoji and regex is not minified properly using default
  // https://github.com/facebook/create-react-app/issues/2488
  ascii_only: true
}
},
// Use multi-process parallel running to improve the build speed
// Default number of concurrent runs: os.cpus().length - 1
parallel: true,
// Enable file caching
cache: true,
sourceMap: shouldUseSourceMap
}),
// This is only used in production mode
new OptimizeCSSAssetsPlugin({
  cssProcessorOptions: {
    parser: safePostCssParser,
```

```

map: shouldUseSourceMap
  ? {
    // `inline: false` forces the sourcemap to be output into a
    // separate file
    inline: false,
    // `annotation: true` appends the sourceMappingURL to the end of
    // the css file, helping the browser find the sourcemap
    annotation: true
  }
  : false
}
}))
],
// Automatically split vendor and commons
// https://twitter.com/wSokra/status/969633336732905474
// https://medium.com/webpack/webpack-4-code-splitting-chunk-graph-and-the-
splitchunks-optimization-be739a861366
splitChunks: {
  chunks: "all",
  name: false
},
// Keep the runtime chunk separated to enable long term caching
// https://twitter.com/wSokra/status/969679223278505985
runtimeChunk: true
},
resolve: {
  // This allows you to set a fallback for where Webpack should look for modules.
  // We placed these paths second because we want `node_modules` to "win"

```

```

// if there are any conflicts. This matches Node resolution mechanism.
// https://github.com/facebook/create-react-app/issues/253
modules: ["node_modules", "src"].concat(
  // It is guaranteed to exist because we tweak it in `env.js`
  process.env.NODE_PATH.split(path.delimiter).filter(Boolean)
),
// These are the reasonable defaults supported by the Node ecosystem.
// We also include JSX as a common component filename extension to support
// some tools, although we do not recommend using it, see:
// https://github.com/facebook/create-react-app/issues/290
// `web` extension prefixes have been added for better support
// for React Native Web.
extensions: paths.moduleFileExtensions
  .map(ext => `.${ext}`)
  .filter(ext => useTypeScript || !ext.includes("ts")),
alias: {
  // Support React Native Web
  // https://www.smashingmagazine.com/2016/08/a-glimpse-into-the-future-with-react-native-for-web/
  "react-native": "react-native-web",
  styles: path.join(paths.appSrc, "styles")
},
plugins: [
  // Adds support for installing with Plug'n'Play, leading to faster installs and adding
  // guards against forgotten dependencies and such.
  PnpWebpackPlugin,
  // Prevents users from importing files from outside of src/ (or node_modules/).
  // This often causes confusion because we only process files within src/ with babel.

```

```

// To fix this, we prevent you from importing files out of src/ -- if you'd like to,
// please link the files into your node_modules/ and let module-resolution kick in.
// Make sure your source files are compiled, as they will not be processed in any way.
new ModuleScopePlugin(paths.appSrc, [paths.appPackageJson])
]
},
resolveLoader: {
  plugins: [
    // Also related to Plug'n'Play, but this time it tells Webpack to load its loaders
    // from the current package.
    PnpWebpackPlugin.moduleLoader(module)
  ]
},
module: {
  strictExportPresence: true,
  rules: [
    // Disable require.ensure as it's not a standard language feature.
    { parser: { requireEnsure: false } },

    // First, run the linter.
    // It's important to do this before Babel processes the JS.
    {
      test: /\.(js|mjs|jsx)$/,
      enforce: "pre",
      use: [
        {
          options: {
            formatter: require.resolve("react-dev-utils/eslintFormatter"),

```

```

    eslintPath: require.resolve("eslint")
  },
  loader: require.resolve("eslint-loader")
}
],
include: paths.appSrc
},
{
  // "oneOf" will traverse all following loaders until one will
  // match the requirements. When no loader matches it will fall
  // back to the "file" loader at the end of the loader list.
  oneOf: [
    // "url" loader works like "file" loader except that it embeds assets
    // smaller than specified limit in bytes as data URLs to avoid requests.
    // A missing `test` is equivalent to a match.
    {
      test: [ /\.bmp$/, /\.gif$/, /\.jpe?g$/, /\.png$/ ],
      loader: require.resolve("url-loader"),
      options: {
        limit: 10000,
        name: "static/media/[name].[hash:8].[ext]"
      }
    },
    // Process application JS with Babel.
    // The preset includes JSX, Flow, TypeScript, and some ESnext features.
    {
      test: /\.(js|mjs|jsx|ts|tsx)$/,
      include: paths.appSrc,

```

```

loader: require.resolve("babel-loader"),
options: {
  customize: require.resolve(
    "babel-preset-react-app/webpack-overrides"
  ),

  plugins: [
    [
      require.resolve("babel-plugin-named-asset-import"),
      {
        loaderMap: {
          svg: {
            ReactComponent: "@svgr/webpack?-svgo,+ref![path]"
          }
        }
      }
    ]
  ],
  // This is a feature of `babel-loader` for webpack (not Babel itself).
  // It enables caching results in ./node_modules/.cache/babel-loader/
  // directory for faster rebuilds.
  cacheDirectory: true,
  cacheCompression: isEnvProduction,
  compact: isEnvProduction
},
// Process any JS outside of the app with Babel.
// Unlike the application JS, we only compile the standard ES features.

```

```

{
  test: /\.(js|mjs)$/,
  exclude: /@babel(?:\|\\{1,2})runtime/,
  loader: require.resolve("babel-loader"),
  options: {
    babelrc: false,
    configFile: false,
    compact: false,
    presets: [
      [
        require.resolve("babel-preset-react-app/dependencies"),
        { helpers: true }
      ]
    ],
    cacheDirectory: true,
    cacheCompression: isEnvProduction,

    // If an error happens in a package, it's possible to be
    // because it was compiled. Thus, we don't want the browser
    // debugger to show the original code. Instead, the code
    // being evaluated would be much more helpful.
    sourceMaps: false
  }
},
// "postcss" loader applies autoprefixer to our CSS.
// "css" loader resolves paths in CSS and adds assets as dependencies.
// "style" loader turns CSS into JS modules that inject <style> tags.
// In production, we use MiniCSSExtractPlugin to extract that CSS

```

```
// to a file, but in development "style" loader enables hot editing
// of CSS.
// By default we support CSS Modules with the extension .module.css
{
  test: cssRegex,
  exclude: cssModuleRegex,
  use: getStyleLoaders({
    importLoaders: 1,
    sourceMap: isEnvProduction && shouldUseSourceMap
  }),
  // Don't consider CSS imports dead code even if the
  // containing package claims to have no side effects.
  // Remove this when webpack adds a warning or an error for this.
  // See https://github.com/webpack/webpack/issues/6571
  sideEffects: true
},
// Adds support for CSS Modules (https://github.com/css-modules/css-modules)
// using the extension .module.css
{
  test: cssModuleRegex,
  use: getStyleLoaders({
    importLoaders: 1,
    sourceMap: isEnvProduction && shouldUseSourceMap,
    modules: true,
    getLocalIdent: getCSSModuleLocalIdent
  })
},
// Opt-in support for SASS (using .scss or .sass extensions).
```

```
// By default we support SASS Modules with the
// extensions .module.scss or .module.sass
{
  test: sassRegex,
  exclude: sassModuleRegex,
  use: getStyleLoaders(
    {
      importLoaders: 2,
      sourceMap: isEnvProduction && shouldUseSourceMap
    },
    "sass-loader"
  ),
  // Don't consider CSS imports dead code even if the
  // containing package claims to have no side effects.
  // Remove this when webpack adds a warning or an error for this.
  // See https://github.com/webpack/webpack/issues/6571
  sideEffects: true
},
// Adds support for CSS Modules, but using SASS
// using the extension .module.scss or .module.sass
{
  test: sassModuleRegex,
  use: getStyleLoaders(
    {
      importLoaders: 2,
      sourceMap: isEnvProduction && shouldUseSourceMap,
      modules: true,
      getLocalIdent: getCSSModuleLocalIdent
```

```

    },
    "sass-loader"
  )
},
// "file" loader makes sure those assets get served by WebpackDevServer.
// When you `import` an asset, you get its (virtual) filename.
// In production, they would get copied to the `build` folder.
// This loader doesn't use a "test" so it will catch all modules
// that fall through the other loaders.
{
  loader: require.resolve("file-loader"),
  // Exclude `js` files to keep "css" loader working as it injects
  // its runtime that would otherwise be processed through "file" loader.
  // Also exclude `html` and `json` extensions so they get processed
  // by webpacks internal loaders.
  exclude: [/\.(js|mjs|jsx|ts|tsx)$/, /\.html$/, /\.json$/],
  options: {
    name: "static/media/[name].[hash:8].[ext]"
  }
}
// ** STOP ** Are you adding a new loader?
// Make sure to add the new loader(s) before the "file" loader.
]
}
]
},
plugins: [
  // Generates an `index.html` file with the <script> injected.

```

```

new HtmlWebpackPlugin(
  Object.assign(
    {},
    {
      inject: true,
      template: paths.appHtml
    },
    isEnvProduction
      ? {
          minify: {
            removeComments: true,
            collapseWhitespace: true,
            removeRedundantAttributes: true,
            useShortDoctype: true,
            removeEmptyAttributes: true,
            removeStyleLinkTypeAttributes: true,
            keepClosingSlash: true,
            minifyJS: true,
            minifyCSS: true,
            minifyURLs: true
          }
        }
      : undefined
  )
),
// Inlines the webpack runtime script. This script is too small to warrant
// a network request.
isEnvProduction &&

```

```

shouldInlineRuntimeChunk &&
  new InlineChunkHtmlPlugin(HtmlWebpackPlugin, [/runtime~.+[.]js/]),
// Makes some environment variables available in index.html.
// The public URL is available as %PUBLIC_URL% in index.html, e.g.:
// <link rel="shortcut icon" href="%PUBLIC_URL%/favicon.ico">
// In production, it will be an empty string unless you specify "homepage"
// in `package.json`, in which case it will be the pathname of that URL.
// In development, this will be an empty string.
new InterpolateHtmlPlugin(HtmlWebpackPlugin, env.raw),
// This gives some necessary context to module not found errors, such as
// the requesting resource.
new ModuleNotFoundPlugin(paths.appPath),
// Makes some environment variables available to the JS code, for example:
// if (process.env.NODE_ENV === 'production') { ... }. See `./env.js`.
// It is absolutely essential that NODE_ENV is set to production
// during a production build.
// Otherwise React will be compiled in the very slow development mode.
new webpack.DefinePlugin(env.stringified),
// This is necessary to emit hot updates (currently CSS only):
isEnvDevelopment && new webpack.HotModuleReplacementPlugin(),
// Watcher doesn't work well if you mistype casing in a path so we use
// a plugin that prints an error when you attempt to do this.
// See https://github.com/facebook/create-react-app/issues/240
isEnvDevelopment && new CaseSensitivePathsPlugin(),
// If you require a missing module and then `npm install` it, you still have
// to restart the development server for Webpack to discover it. This plugin
// makes the discovery automatic so you don't have to restart.
// See https://github.com/facebook/create-react-app/issues/186

```

```

isEnvDevelopment &&
  new WatchMissingNodeModulesPlugin(paths.appNodeModules),
isEnvProduction &&
  new MiniCssExtractPlugin({
    // Options similar to the same options in webpackOptions.output
    // both options are optional
    filename: "static/css/[name].[contenthash:8].css",
    chunkFilename: "static/css/[name].[contenthash:8].chunk.css"
  }),
  // Generate a manifest file which contains a mapping of all asset filenames
  // to their corresponding output file so that tools can pick it up without
  // having to parse `index.html`.
  new ManifestPlugin({
    fileName: "asset-manifest.json",
    publicPath: publicPath
  }),
  // Moment.js is an extremely popular library that bundles large locale files
  // by default due to how Webpack interprets its code. This is a practical
  // solution that requires the user to opt into importing specific locales.
  // https://github.com/jmblog/how-to-optimize-momentjs-with-webpack
  // You can remove this if you don't use Moment.js:
  new webpack.IgnorePlugin(/^\.\/locale$/, /moment$/),
  // Generate a service worker script that will precache, and keep up to date,
  // the HTML & assets that are part of the Webpack build.
isEnvProduction &&
  new WorkboxWebpackPlugin.GenerateSW({
    clientsClaim: true,
    exclude: [/\.map$/, /asset-manifest\.json$/],
  })

```

```

importWorkboxFrom: "cdn",
navigateFallback: publicUrl + "/index.html",
navigateFallbackBlacklist: [
  // Exclude URLs starting with /_, as they're likely an API call
  new RegExp("^/_"),
  // Exclude URLs containing a dot, as they're likely a resource in
  // public/ and not a SPA route
  new RegExp("/[^\./]+\.[^\./]+$")
]
}),
// TypeScript type checking
useTypeScript &&
new ForkTsCheckerWebpackPlugin({
  typescript: resolve.sync("typescript", {
    basedir: paths.appNodeModules
  }),
  async: isEnvDevelopment,
  useTypescriptIncrementalApi: true,
  checkSyntacticErrors: true,
  tsconfig: paths.appTsConfig,
  reportFiles: [
    "**",
    "!**/*.json",
    "!**/__tests__/**",
    "!**/?(*.)(spec|test).*",
    "!**/src/setupProxy.*",
    "!**/src/setupTests.*"
  ],

```

```
    watch: paths.appSrc,
    silent: true,
    // The formatter is invoked directly in WebpackDevServerUtils during development
    formatter: isEnvProduction ? typescriptFormatter : undefined
  })
].filter(Boolean),
// Some libraries import Node modules but don't use them in the browser.
// Tell Webpack to provide empty mocks for them so importing them works.
node: {
  module: "empty",
  dgram: "empty",
  dns: "mock",
  fs: "empty",
  net: "empty",
  tls: "empty",
  child_process: "empty"
},
// Turn off performance processing because we utilize
// our own hints via the FileSizeReporter
performance: false
};
};
```

## Додаток Г

КОПІЇ ПРЕЗЕНТАЦІЇ

Міністерство освіти і науки України  
Харківський національний університет  
радіоелектроніки

1

КВАЛІФІКАЦІЙНА РОБОТА НА ТЕМУ:  
«Розробка діагностичного додатку для контролю якості зору очей»

Виконав:  
ст. гр. ІПРу-21-1  
Дейчман Я.Б

Керівник роботи:  
Професор Кузьомін О.Я

Мета, об'єкт, предмет  
кваліфікаційної роботи

2

Мета даної кваліфікаційної роботи полягає в розробці додатку для мобільних пристроїв за допомогою якого користувачі зможуть самостійно протестувати стан та якість свого зору.

Об'єктом досліджень кваліфікаційної роботи є діагностичний додаток для контролю якості зору очей.

Предметом є програмні методи створення мобільного додатку для тестування і контролю якості зору очей.

3

## Постановка завдання

Розробити серверну та клієнтську частини додатку для контролю якості зору очей. Серверна частина повинна являти собою базу даних, розробленої для платформи СУБД mysql. Клієнтська частина має надавати користувачу доступ до тестувань очей і результатів цих тестувань. Операційна система повинна бути Android 8.0 або новішою, інтегроване середовище розробки – microsoft visual studio code 2019 або пізнішої версії, мова програмування – React.

4

## Аналіз предметної області. Актуальність теми

Близько 80% інформації про навколишній світ, яка надходить до нашого мозку, ми отримуємо через зір. Зорова кора, розташована в задній частині мозку, є основним центром для отримання, інтеграції та обробки візуальної інформації від сітківки ока. Дослідження показали значний зв'язок між мигдалеподібним тілом і зоровою корою. Останніми роками лікарі відзначають загальне погіршення гостроти зору не лише серед дорослих і літніх людей, а й серед молоді та підлітків. Важливо не ігнорувати перші ознаки погіршення зору, такі як підвищена втомлюваність очей, почервоніння та подразнення повік, і своєчасно звертатися до офтальмолога. Перевірка зору за допомогою онлайн-сервісів або додатків допоможе комфортно та просто стежити за станом зору, а у разі його погіршення — своєчасно звернутися до лікаря-офтальмолога для більш точної консультації.

5

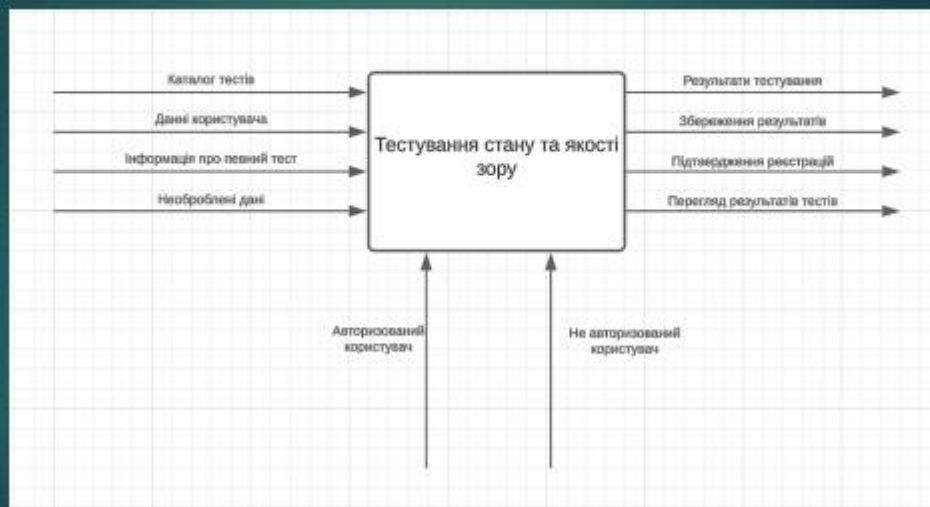
## Постановка задачі на розробку компонентів системи

Головною задачею надання користувача можливості контролювати стан свого зору за допомогою різноманітних тестів які йому пропонує розроблений додаток. Також важливо реалізувати авторизацію та реєстрацію користувачів для того щоб користувач зміг переглянути результати минулих тестувань. Крім того, важливо розглянути проектування зручного та інтуїтивно зрозумілого користувацького інтерфейсу.

6

## Розробка вимог. Концептуальна діаграма

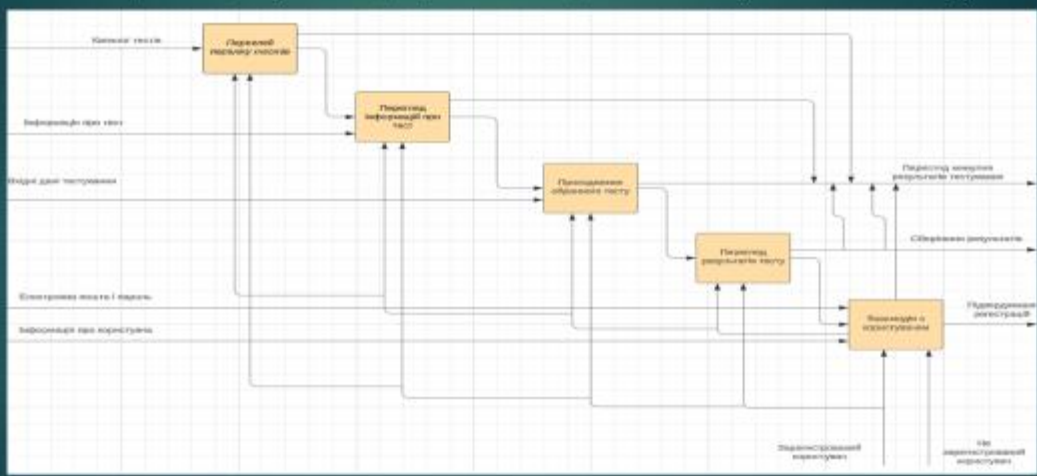
Концептуальна діаграма "Контроль та тестування якості зору очей"



# Розробка вимог. Декомпозиція концептуальної діаграми нульового рівня

Декомпозиція концептуальної діаграми "Контроль та тестування якості зору очей"

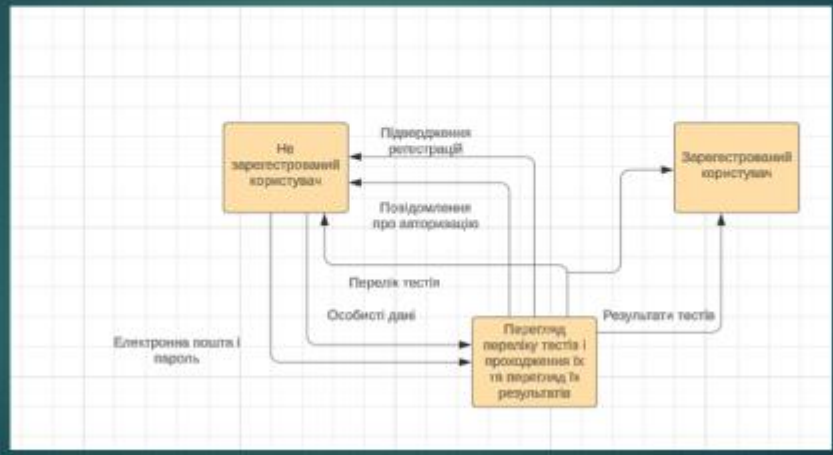
7



# Розробка вимог. Контекстна діаграма DFD

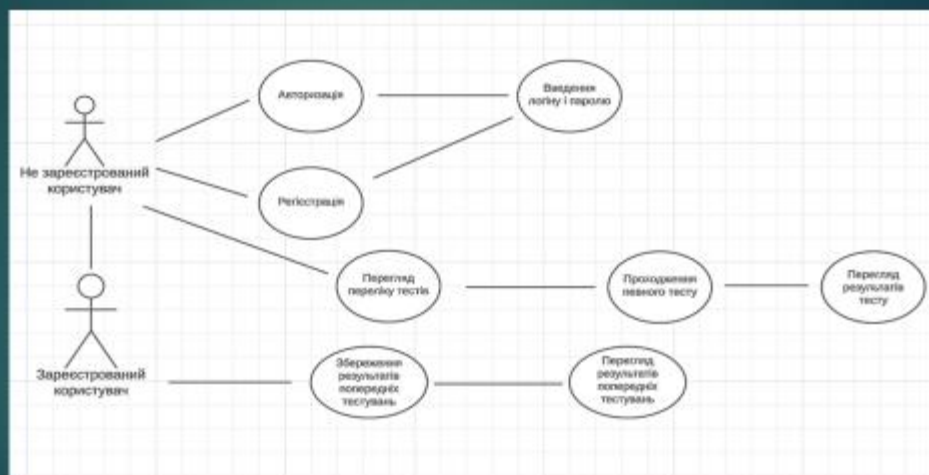
8

Контекстна діаграма DFD "Контроль та тестування якості зору очей"



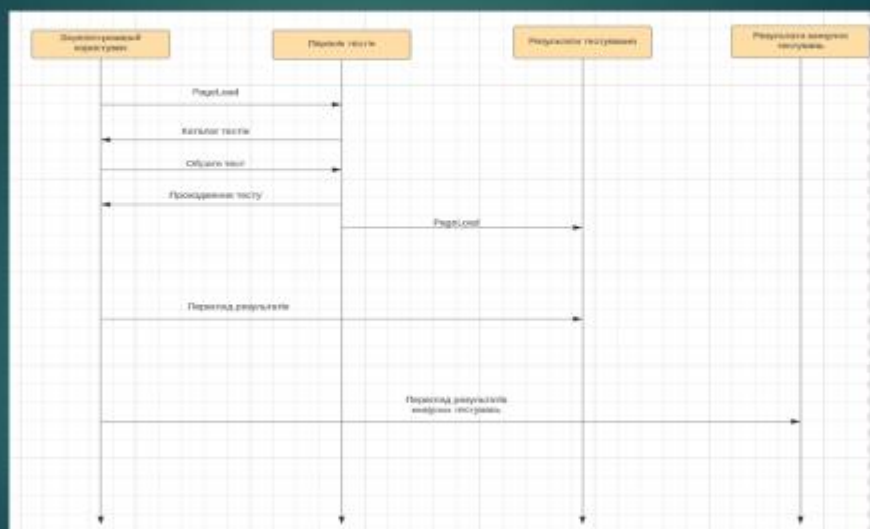
## Визначення вимог. Діаграма варіантів використання

9



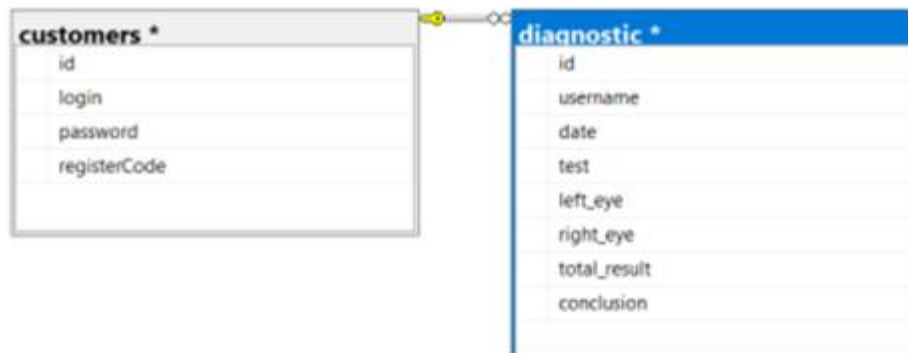
## Визначення вимог. Діаграма послідовності дій для прецеденту

10



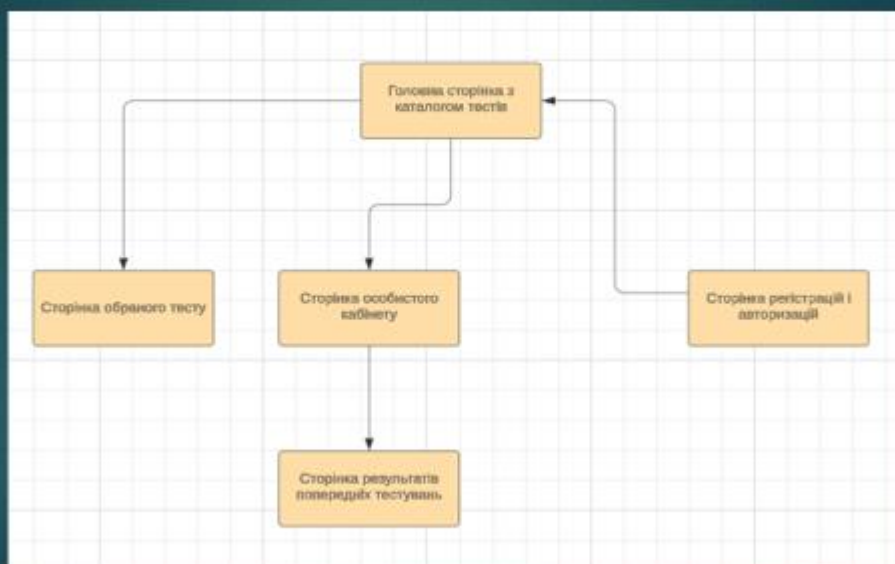
## Логічна модель бази даних додатку

11



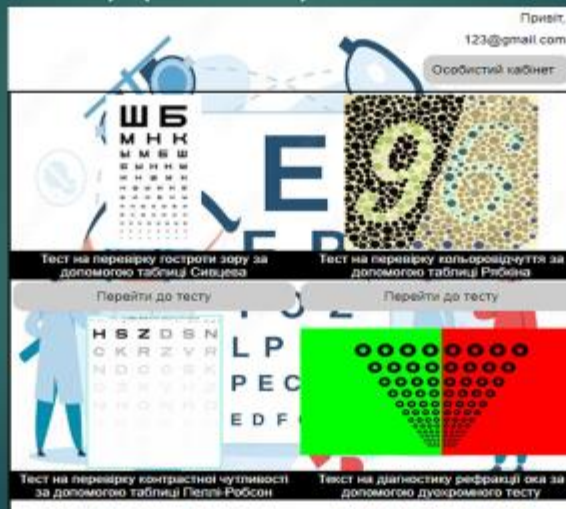
## Карта сторінок додатку

12



## Скріншот інтерфейсу з реалізованою функцією перегляду переліку тестів і вибору для проходження

13



## Програмний код реалізації функції перегляду переліку тестів і вибору для проходження

14

```

ident: "postcss",
plugins: () => [
  require("postcss-flexbugs-fixes"),
  require("postcss-preset-env")({
    autoprefixer: {
      flexbox: "no-2009"
    },
    stage: 3
  })
],
sourceMap: isEnvProduction && shouldUseSourceMap
}
].filter(Boolean);
if (preProcessor) {
  loaders.push({
    loader: require.resolve(preProcessor),
    options: {
      sourceMap: isEnvProduction && shouldUseSourceMap
    }
  });
}
return loaders;
};

return {
  mode: isEnvProduction ? "production" : isEnvDevelopment && "development",

```



## Скріншот інтерфейсу з реалізованою функцією проходження вибраного тесту

17



## Програмний код реалізації функції проходження вибраного тесту

18

```
test: /\.?(js|mjs|jsx|ts|tsx)$/,
include: paths.appSrc,
loader: require.resolve("babel-loader"),
options: {
  customize: require.resolve(
    "babel-preset-react-app/webpack-overrides"
  ),
  plugins: [
    [
      require.resolve("babel-plugin-named-asset-import"),
      {
        loaderMap: {
          svg: {
            ReactComponent: "@svgr/webpack?-svgo,+ref![path]"
          }
        }
      }
    ]
  ]
}
```

## Скріншот інтерфейсу з реалізованою функцією відображення результатів тестування

19



## Програмний код реалізації функції відображення результатів тестування

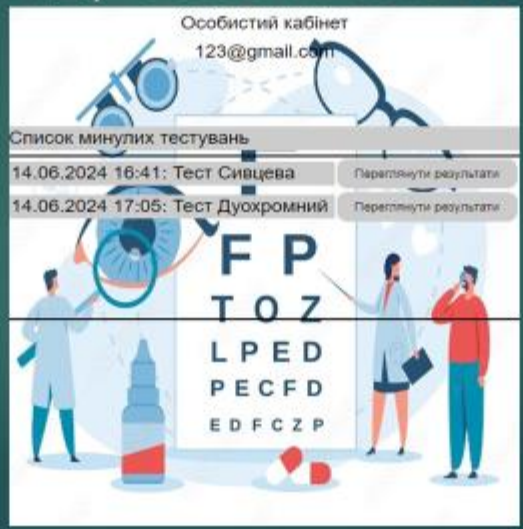
20

```
const REACT_APP = /^REACT_APP_/i;

function getClientEnvironment(publicUrl) {
  const raw = Object.keys(process.env)
    .filter(key => REACT_APP.test(key))
    .reduce(
      (env, key) => {
        env[key] = process.env[key];
        return env;
      },
      {}
    );
}
```

# Скріншот інтерфейсу з реалізованою функцією перегляду особистого кабінету

21



# Програмний код реалізації функції перегляду особистого кабінету

22

```

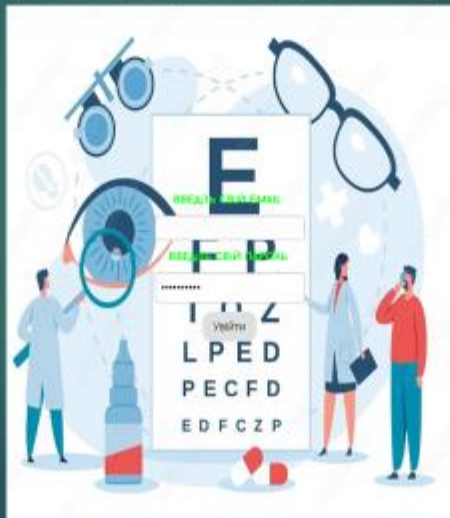
isEnvProduction 55
new WorkboxWebpackPlugin.GenerateSW({
  clientsClaim: true,
  exclude: [/\_map$/, /asset-manifest\.json$/],
  importWorkboxFrom: "cdn",
  navigateFallback: publicUrl + "/index.html",
  navigateFallbackBlacklist: [
    // Exclude URLs starting with /_ as they're likely an API call
    new RegExp("^/_/"),
    // Exclude URLs containing a dot, as they're likely a resource in
    // public/ and not a SPA route
    new RegExp("/[^/]+\.[^/]+$")
  ]
}),
// TypeScript type checking
useTypeScript 55
new ForkTsCheckerWebpackPlugin({
  typescript: resolve.sync("typescript", {
    basedir: paths.appNodeModules
  })
})

```



## Скріншот інтерфейсу з реалізованою функцією авторизації користувача

25



## Програмний код реалізації функції авторизації користувача

26

```

<li>
  <Link to="/login">login</Link>
</li>
<li>
  <Link to="/reset">Reset</Link>
</li>
<li>
  <Link to="/dashboard">Dashboard</Link>
</li>
</ul>
</nav>

<Routes>
  <Route path="/" element={Home} />
  <Route path="/login" element={Login} />
  <Route path="/reset" element={PasswordReset} />
  <Route path="/dashboard" element={DefaultLayout} />
</Routes>

```

# Інструкція користувача

27

## Інструкція користувача

Контроль якості зору виконаний для дівчаток віком 8 років і ранішого виконання буде вважатися проблемою. Для цього встановити додатково якість зору та додатково діаметр окуляра та окуляра.

### Інструкції з контролю якості зору:

#### 1. Підготовка користувача

Виконати якість зору згідно з протоколом, щоб перевірити зору на наявність зору.

#### 2. Технічні вимоги

- **Технічні вимоги:**
  - Використовувати тільки дитячі окуляри.
  - Застосувати окуляри для дітей, якщо на них користувач.
  - Використовувати окуляри на дитячій висоті, але не менше 14-15 сантиметрів.
  - Використовувати окуляри, а не інші окуляри, якщо в них є окуляри.
  - Використовувати окуляри, а не інші окуляри, якщо в них є окуляри.
  - Використовувати окуляри, а не інші окуляри, якщо в них є окуляри.

#### 3. Підготовка користувача

Виконати якість зору згідно з протоколом, щоб перевірити зору на наявність зору.

- **Підготовка користувача:**
  - Використовувати тільки дитячі окуляри.
  - Використовувати окуляри, а не інші окуляри, якщо в них є окуляри.
  - Використовувати окуляри, а не інші окуляри, якщо в них є окуляри.
  - Використовувати окуляри, а не інші окуляри, якщо в них є окуляри.
  - Використовувати окуляри, а не інші окуляри, якщо в них є окуляри.

#### 4. Підготовка користувача

Виконати якість зору згідно з протоколом, щоб перевірити зору на наявність зору.

- **Контроль якості зору:**
  - Використовувати тільки дитячі окуляри.
  - Використовувати окуляри, а не інші окуляри, якщо в них є окуляри.
  - Використовувати окуляри, а не інші окуляри, якщо в них є окуляри.

## Огляд функцій та налаштувань

### 4. Налаштування та налаштування

Виконати якість зору згідно з протоколом, щоб перевірити зору на наявність зору.

- **Налаштування користувача:**
  - Використовувати тільки дитячі окуляри.
  - Використовувати окуляри, а не інші окуляри, якщо в них є окуляри.
  - Використовувати окуляри, а не інші окуляри, якщо в них є окуляри.
- **Налаштування користувача:**
  - Використовувати тільки дитячі окуляри.
  - Використовувати окуляри, а не інші окуляри, якщо в них є окуляри.
  - Використовувати окуляри, а не інші окуляри, якщо в них є окуляри.
- **Налаштування користувача:**
  - Використовувати тільки дитячі окуляри.
  - Використовувати окуляри, а не інші окуляри, якщо в них є окуляри.
  - Використовувати окуляри, а не інші окуляри, якщо в них є окуляри.
- **Налаштування користувача:**
  - Використовувати тільки дитячі окуляри.
  - Використовувати окуляри, а не інші окуляри, якщо в них є окуляри.
  - Використовувати окуляри, а не інші окуляри, якщо в них є окуляри.

### 5. Налаштування

Виконати якість зору згідно з протоколом, щоб перевірити зору на наявність зору.

- **Налаштування користувача:**
  - Використовувати тільки дитячі окуляри.
  - Використовувати окуляри, а не інші окуляри, якщо в них є окуляри.
  - Використовувати окуляри, а не інші окуляри, якщо в них є окуляри.
- **Налаштування користувача:**
  - Використовувати тільки дитячі окуляри.
  - Використовувати окуляри, а не інші окуляри, якщо в них є окуляри.
  - Використовувати окуляри, а не інші окуляри, якщо в них є окуляри.

Виконати якість зору згідно з протоколом, щоб перевірити зору на наявність зору.

# Висновки

28

Під час виконання кваліфікаційної роботи був створений мобільний додаток для контролю на тестування якості зору очей. В рамках розробки було виконано наступні етапи:

Визначено тип системи, розроблені системні та функціональні вимоги, моделі потоків даних системи, UML-діаграми, проведено логічне та фізичне моделювання даних, реалізована база даних системи. Виконана робота надає базу для подальшого вдосконалення даного додатку.

29

Дякую за увагу!

30

Міністерство освіти і науки України  
Харківський національний університет  
радіоелектроніки

КВАЛІФІКАЦІЙНА РОБОТА НА ТЕМУ:  
«Розробка діагностичного додатку для контролю якості  
зору очей»

Виконав:  
ст. гр. ІПРу-21-1  
Дейчман Я.Б

Керівник роботи:  
Професор Кузьомін О.Я

## Додаток Д

## ВІДОМІСТЬ КВАЛІФІКАЦІЙНОЇ РОБОТИ

