|  |  |
| --- | --- |
|  | **ИКОНОМИЧЕСКИ УНИВЕРСИТЕТ - ВАРНА**  **КАТЕДРА „ИНФОРМАТИКА”** |

*х.ас.докт. Йордан Иванов Йорданов*

Проект на тема:

**Модел на Семпъл онлайн магазин базиран на облачната платформа Microsoft Azure**

по дисциплина „*Езици за програмиране*“

**Варна 2021**

**Съдържание**

Въведение 3

1. Подходи за дизайн на системата 4

2. Модели на подсистемите в онлайн магазина 8

3. Архитектура на софтуерното внедряване в облачна среда 10

Заключение 15

Използвана литература 15

# Въведение

За кратко време облачните технологии се превърнаха в водеща тенденция в софтуерната индустрия. Те представят нов начин, по който да се водим и мислим, за изграждането на големи, сложни системи, подход, който се възползва изцяло от съвременните практики за разработка на високо-качествен софтуер и инфраструктура. Предлага промени в начина, по който проектираме, интегрираме и внедряваме системите. Облачно базираните системи са проектирани да приемат бързо промените, да обслужват голям мащаб от хора и да бъдат устойчиви на всякакъв вид натоварване или хакерски атаки.

Cloud Native Computing Foundation[[1]](#footnote-1) предоставя официално определение:

Технологиите, базирани на облак, дават възможност на организациите да създават и изпълняват мащабируеми приложения в модерни, динамични среди като публични, частни и хибридни облаци, чрез мрежи от услуги и микроуслуги. Качества на системите са устойчивост, висока наличност и достъпност, мащабируемост и управляемост, които са критични преимущества за много от бизнес единиците. Автоматизацията на тези процеси позволява на инженерите да правят промени, с голямо въздействие, но с минимални усилия.

Приложенията стават все по-сложни, като изискванията от потребителите са все повече и повече, главно насочени към бърза реакция, иновативни функции и нулев престой. Проблеми с производителността, повтарящи се грешки и базови хакерски атаки вече не са приемливи. Тези предиства поставят бизнеса на една стъпка пред конкурентите. Облачно базираните системи се свързват главно с бързина. Бизнес системите се развиват от способностите на бизнеса да бъдат инструменти за стратегическа трансформация, като ускорят растежа на компанията. Незабавното пускане на иновативните идеи на пазара е от критична тема за всички модерни компании. Нека разгледаме два примера, които са приложили тези техники:

- Netflix има над 600 услуги в производствена среда. Стотици пъти на ден се изпълняват нови внедрявания и разгръщания на същестуваши.

- Uber има над 1000 услуги в производствена среда. Разгръщат се няколко хиляди пъти всяка седмица.

Както може да видим тези две компании са базирани на системи, които се състоят от стотици независими микроуслуги. Този архитектурен стил им позволява бързо да реагират на пазарните условия като постоянно да актуализират малки, но важни области. Скоростта и пъргавината на облачния носител се дължат на редица фактори, като на първо място е инфраструктурата на изчислителните ресурси. По примери и указания на водещи експерти от общността, нека разгледаме характеристики и изисквания на функционален облачен продукт, демонстриращ използването на .NET, Docker, Kubernetes в облачната среда на Microsoft Azure за да се създаде онлайн магазин.

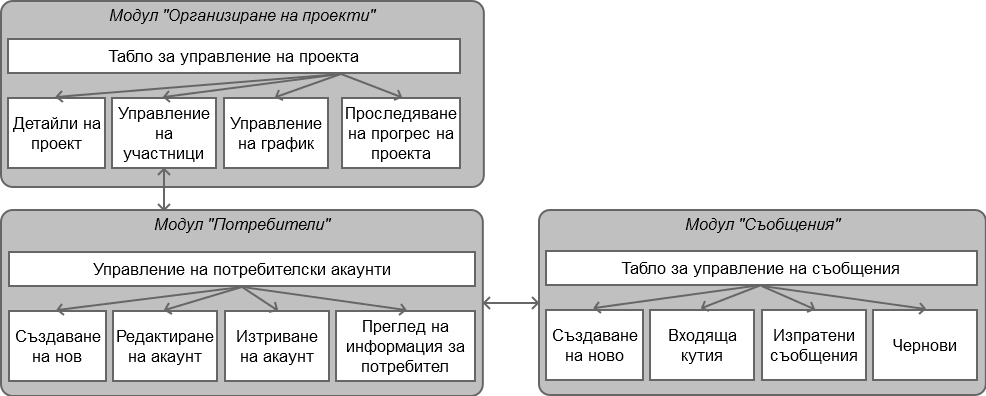
# Подходи за дизайн на системата

Достъп до уеб базираното приложение имат 4 групи потребители: Администратор (А), Мениджър на проект (МП), Анализатор на ползваемостта (АП) и Тестер (Т). За всяка група се поддържа различен интерфейс. Както препоръчват някои автори[[2]](#footnote-2), страниците, които се използват в тези интерфейси, е необходимо да бъдат групирани в модули, за да се ограничи достъпът на някои от групите до определени функционалности на приложението.

Приложението е изградено от следните основни модули:

* *Организиране на проекти* – модулът е отговорен за създаване на нови проекти, управление на активните, проследяване на резултатите от вече приключилите проекти и премахване на проекти. Включва функционалности по управление на основната информация за проект, на участниците в проекта, на графика на проекта, на потребителските сесии и извеждане на резултатите от тестовете. Основната информация, която се съхранява за всеки проект, е наименование, начална и крайна дата, описание, снимка и статус. Поддържаните графични файлови формати са jpg и png, а максимално допустимият размер на файла е 1МВ. Ако не се публикува снимка за проекта, системата показва такава по подразбиране. Всеки проект може да има три статуса – В процес на изпълнение, Преустановен, Приключен. При настъпване на началната дата на даден проект автоматично се променя статусът му на „В процес на изпълнение“ и става видим за потребителите на системата, които са асоциирани като участници. При този статус не могат да се извършват промени. За да се извършват промени по графика на изпълнение и / или по потребителските сесии, както и при временно спиране на провеждането на проекта, статусът трябва да бъде променен на „Преустановен“. Това може да се направи само от Администратор или Мениджър. Проектът придобива автоматично статус „Приключен“ след крайната му дата.
* *Потребители* – модулът включва функционалности по създаване на нови и изтриване на съществуващи потребители, управление на профилната информация и индивидуалните настройки на системата. За всеки потребител в системата се съхраняват потребителско име, парола, роля (потребителска група), email, телефонен номер, местоположение и снимка. Поддържаните графични файлови формати са jpg и png, а максимално допустимият размер на файла е 1МВ. Ако потребителят не промени профилната си снимка, системата показва такава по подразбиране, съответстваща на ролята му.
* *Съобщения* – модулът е отговорен за обмена на съобщения между потребителите на системата. Могат да се проследяват изпратени и получени съобщения, както и запазените чернови. Потребителите могат да изпращат съобщения с прикачени файлове. Позволените файлови разширения са pdf, doc, docx, xls, xlsx, avi, mp4, mp3, jpg, png. Максимално позволеният размер на файла е 5 МВ.

Фиг. 1 илюстрира основните модули на уеб базираното приложение и връзките помежду им. Същите са част и от цялостната система за оценяване на ползваемостта на мобилни приложения.



**Фиг. 1. Модули на уеб базираното приложение**

Администраторът използва системата с неограничени права. Единствено той може да създава и изтрива проекти, да регистрира и изтрива потребители.

Мениджърът на проект е потребителят, който администрира проекти. Един проект може да се управлява само от един мениджър, докато един мениджър може да управлява повече от един проект. Има почти същите права като администратора на системата, с изключение на това, че не може да създава и изтрива проекти, да регистрира и изтрива потребители.

Анализаторът на ползваемостта е потребител на системата, чиято роля е да проследява резултатите от изпълнение на тестовите сесии. Той има ограничени права по отношение на управление на проекта. Може да променя профилните си данни, да работи с модул „Съобщения“, да задава индивидуални настройки на приложението, да управлява потребителски сесии и да проследява резултатите от изпълнението им.

Администраторът, мениджърът на проекти и анализаторът не могат да вземат участие в потребителски сесии при тестване на ползваемостта. Тази роля се изпълнява от тестера.

Тестерът е потребител, чиито права за достъп до приложението са най-ограничени. Той може да променя профилните си данни, да работи с модул „Съобщения“, да задава индивидуални настройки на приложението и да участва в потребителски сесии.

Таблица 1 отразява правата на съответните групи потребители за достъп до модулите на приложението.

**Таблица 1**

**Права за достъп до системата на отделните потребителски групи**

| **Модул** | **Т** | **АП** | **МП** | **А** |
| --- | --- | --- | --- | --- |
| **Организиране на проекти** | | | | | |
| 1. Създаване на нов проект |  |  |  | Kinjrykiq |
| 1. Управление на проект |  |  | Kinjrykiq | Kinjrykiq |
| * 1. Управление на основната информация за проект |  |  | Kinjrykiq | Kinjrykiq |
| * 1. Управление на участниците в проекта |  |  | Kinjrykiq | Kinjrykiq |
| * 1. Управление на графика на проекта |  |  | Kinjrykiq | Kinjrykiq |
| * 1. Управление на потребителските сесии |  | Kinjrykiq | Kinjrykiq | Kinjrykiq |
| * + 1. Управление на интервюта |  | Kinjrykiq | Kinjrykiq | Kinjrykiq |
| * + 1. Управление на тестове на ползваемост |  | Kinjrykiq | Kinjrykiq | Kinjrykiq |
| * + 1. Достъп до Бележник |  | Kinjrykiq | Kinjrykiq | Kinjrykiq |
| 1. Проследяване на резултати от изпълнение на проект |  | Kinjrykiq | Kinjrykiq | Kinjrykiq |
| 1. Участие в потребителски сесии | Kinjrykiq |  |  |  |
| 1. Преустановяване на проект |  |  | Kinjrykiq | Kinjrykiq |
| 1. Изтриване на проект |  |  |  | Kinjrykiq |
| **Потребители** | | | | | |
| 1. Създаване на нов потребител |  |  |  | C:\Users\Radka\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Kinjrykiq.png |
| 1. Промяна на профилни данни | C:\Users\Radka\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Kinjrykiq.png | C:\Users\Radka\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Kinjrykiq.png | C:\Users\Radka\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Kinjrykiq.png | C:\Users\Radka\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Kinjrykiq.png |
| 1. Изтриване на потребител |  |  |  | C:\Users\Radka\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Kinjrykiq.png |
| 1. Задаване на индивидуални настройки на приложението | C:\Users\Radka\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Kinjrykiq.png | C:\Users\Radka\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Kinjrykiq.png | C:\Users\Radka\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Kinjrykiq.png | C:\Users\Radka\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Kinjrykiq.png |
| **Съобщения** | | | | | |
| 1. Изпращане на ново приложение | C:\Users\Radka\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Kinjrykiq.png | C:\Users\Radka\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Kinjrykiq.png | C:\Users\Radka\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Kinjrykiq.png | C:\Users\Radka\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Kinjrykiq.png |
| * 1. Изпращане на прикачени файлове | C:\Users\Radka\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Kinjrykiq.png | C:\Users\Radka\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Kinjrykiq.png | C:\Users\Radka\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Kinjrykiq.png | C:\Users\Radka\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Kinjrykiq.png |
| 1. Проследяване на изпратени съобщения | C:\Users\Radka\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Kinjrykiq.png | C:\Users\Radka\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Kinjrykiq.png | C:\Users\Radka\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Kinjrykiq.png | C:\Users\Radka\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Kinjrykiq.png |
| 1. Проследяване на получени съобщения | C:\Users\Radka\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Kinjrykiq.png | C:\Users\Radka\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Kinjrykiq.png | C:\Users\Radka\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Kinjrykiq.png | C:\Users\Radka\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Kinjrykiq.png |
| 1. Управление на чернови | C:\Users\Radka\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Kinjrykiq.png | C:\Users\Radka\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Kinjrykiq.png | C:\Users\Radka\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Kinjrykiq.png | C:\Users\Radka\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Kinjrykiq.png |
| * 1. Редактиране на чернова |  |  |  |  |
| * 1. Изпращане на чернова |  |  |  |  |
| * 1. Изтриване на чернова |  |  |  |  |

Входът в системата за всички потребители се осъществява след въвеждане на потребителско име и парола. Всеки потребител може да проследява активността на профила си. Приложението показва история на влизанията.

Необходимо условие за осигуряване на универсален достъп до приложението е потребителският му интерфейс да бъде адаптивен. По този начин ще се предостави възможност на потребителите да работят с него през мобилните си устройства.

Трябва да се има предвид, че се планира в бъдеще следва да бъдат приложени технологии за събиране и обработка на биометрични данни, поради факта, че приложението ще функционира като уеб базирано приложение за оценяване на ползваемостта на мобилни приложения. Ето защо, е необходимо при подборът на технологии за разработката му да се вземат предвид и тези особености.

# Модели на подсистемите в онлайн магазина

Изборът на конкретни технологии за разработване на уеб базираното приложение е съобразен с предназначението му и перспективите за бъдещото му развитие.

Важни акценти в изискванията към приложението са:

* Приложението трябва да бъде уеб базирано;
* Дизайнът трябва да е адаптивен;
* Приложението трябва да бъде изградено на модулен принцип;
* В бъдеще ще бъдат приложени технологии за събиране и обработка на биометрични данни.

Като подходяща за целта комбинация от технологии и софтуерни средства може да се определи:

* *Език за програмиране*: Java;
* *Технологии и софтуерни средства за разработване на потребителския интерфейс*: HTML 5, CSS 3, JavaScript, библиотека jQuery, Bootstrap;
* *Система за управление на бази данни*: MySQL;
* *Уеб сървър*: Apache Tomcat.

Изборът на Java като език за програмиране е основан на главните му предимства, именно:

* Обектноориентиран;
* Платформено независим - лесна преносимост между различни платформи;
* Висока степен на сигурност;
* Възможност за разработване на многонишкови приложения;
* Висока производителност на приложенията;
* Възможност за контрол на правата на потребителя на ниво виртуална машина, др.

За изпълнение на уеб базирано Java приложение е необходим уеб сървър за Java. Примери за такива са Apache Tomcat, Jetty, Apache TomEE, Apache Geronimo и др. В настоящия проект се използва Apache Tomcat като един от най-широко разпространените и използвани в практиката уеб сървъри за изпълнение на Java EE приложения.

Избраната комбинация от технологии и софтуерни средства за разработване на потребителския интерфейс може да се определи като широко използвана към настоящия момент. Основните им предимства се изразяват в:

* Спестяване на време в процеса на разработка;
* Осигуряване на стандартна работна система, чрез която може да се развие желания модул или приложение;
* Съсредоточаване върху разработването на изискванията, не върху подготовката на инструментариума за разработка на приложението;
* Следват се конвенции в кодирането;
* Разделя се бизнес логиката от потребителския интерфейс, като по този начин кода се запазва изчистен и се предоставя възможност за разширяемост на разработката.

Основните позитиви, които носи използването на Bootstrap при разработване на потребителския интерфейс на приложението, се свеждат предимно до:

* Интегрира добри практики за адаптивен дизайн, основната сред които е използването на т. нар. „решетъчна система“[[3]](#footnote-3), използвана за подредба на съдържанието. Включени са 12 колони и се поддържат CSS класове за мобилни телефони, таблети, десктоп и големи екрани;
* Прилага подхода “mobile-first” - част от ядрото на Bootstrap са стиловете, ориентирани към разработване на мобилни уеб сайтове;
* Съвместимост с всички съвременни браузъри, сред които Chrome, Firefox, Internet Explorer, Safari и Opera;
* Съвместимост с библиотеката jQuery.

Едно от основните предимства от използване на JavaScript при разработване на потребителския интерфейс на приложението е добавяне на интерактивност – разширяване на функционалността на уеб страниците, включително извършване на някои операции динамично, само от страната на клиента, без да се отправят излишни заявки към сървъра. Например, при валидиране на форми, при филтриране и сортиране на данни, при извеждане на насочващи съобщения, при странициране от страна на клиента и т.н.

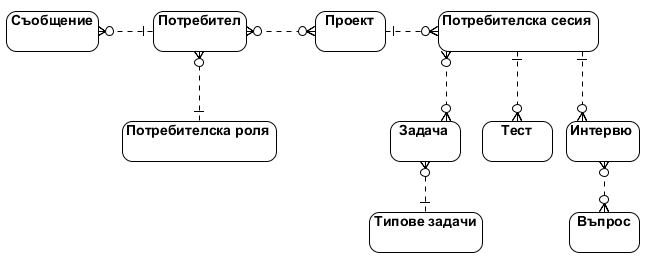
Вземайки предвид изискванията към приложението и избраните за разработката му технологии, то е необходимо като следваща стъпка от проекта да се премине към неговото проектиране. Коректното извършване на тази стъпка намалява грешките по време на разработката му и съответно времето за осъществяване на цялостния проект.

# Архитектура на софтуерното внедряване и поддръжка в облачна среда

Съществен момент в процеса на разработка на приложението е проектирането на базата данни. Добре проектираната база данни ще осигури по-голямо бързодействие и по-добра производителност на приложението. Настоящият проект използва релационна база данни, разработена с инструмента phpMyAdmin.

На база на изискванията към приложението могат да се дефинират следните същности (основните обекти, за които следва да се съхранява информация в базата данни): Потребител, Потребителска роля, Проект, Потребителска сесия, Задача, Тип на задача, Интервю, Въпрос, Тест, Съобщение.

На фиг. 2 е отразен начален E-R модел на базата данни, който служи като основа за разработване на физическата база данни.



**Фиг. 2. Начален E-R модел на базата данни[[4]](#footnote-4)**

Трябва да се има предвид още, че в базата данни ще се съдържат и таблици, в които се съхранява информация за модулите на приложението и правата за достъп на отделните потребителски групи.

Страниците на приложението са разработени, като JSP (Java Server Pages) – поддържат динамично съдържание под формата на Java код, добавяно чрез JSP тагове и комбинирано с HTML тагове. Логиката е разделена от потребителския интерфейс. Разработена е под формата на Java сървлети, организирани в пакети и извиквани в съответните страници на приложението при извършване на конкретни обработки. Създадени са пакети, групиращи сървлети за достъп до базата данни, за управление на входа и регистрациите, за управление на сесиите и бисквитките, за манипулиране с данни и др.

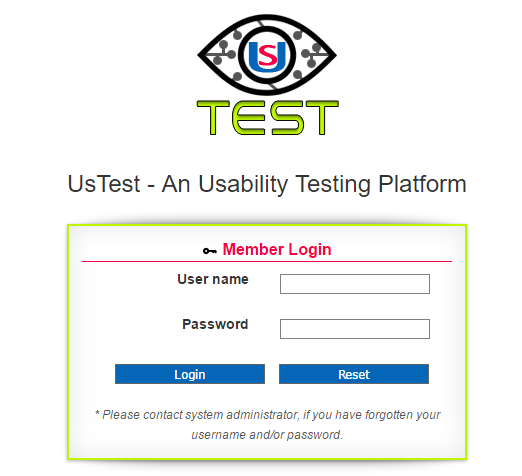
За да се спазят изискванията към приложението, страниците са групирани в модули, за да се ограничи достъпът на различните групи потребители до несвойствени за ролята им в приложението функционалности. Модулният принцип на изграждане на софтуер спомага за по-лесното им развитие, тестване, внедряване и поддържане. От друга страна, по този начин се идентифицират аспекти от приложенията, съдържащи широк кръг от функционални, които се разработват и тестват независимо едни от други. Друго предимство от прилагане на модулния подход е, че той може да направи цялостната архитектура по-гъвкава – ще се осигурят лесно управляеми компоненти, които капсулират специфична функционалност и са интегрирани чрез ясни, но свободно съчетани комуникационни канали. Също така ще се даде възможност за лесна разширяемост в бъдеще.

Информацията за организацията на страниците в модули и достъпът на потребителите до конкретните модули се съхранява в базата данни в специално създадени за целта таблици.

**Проектиране на потребителския интерфейс на приложението**

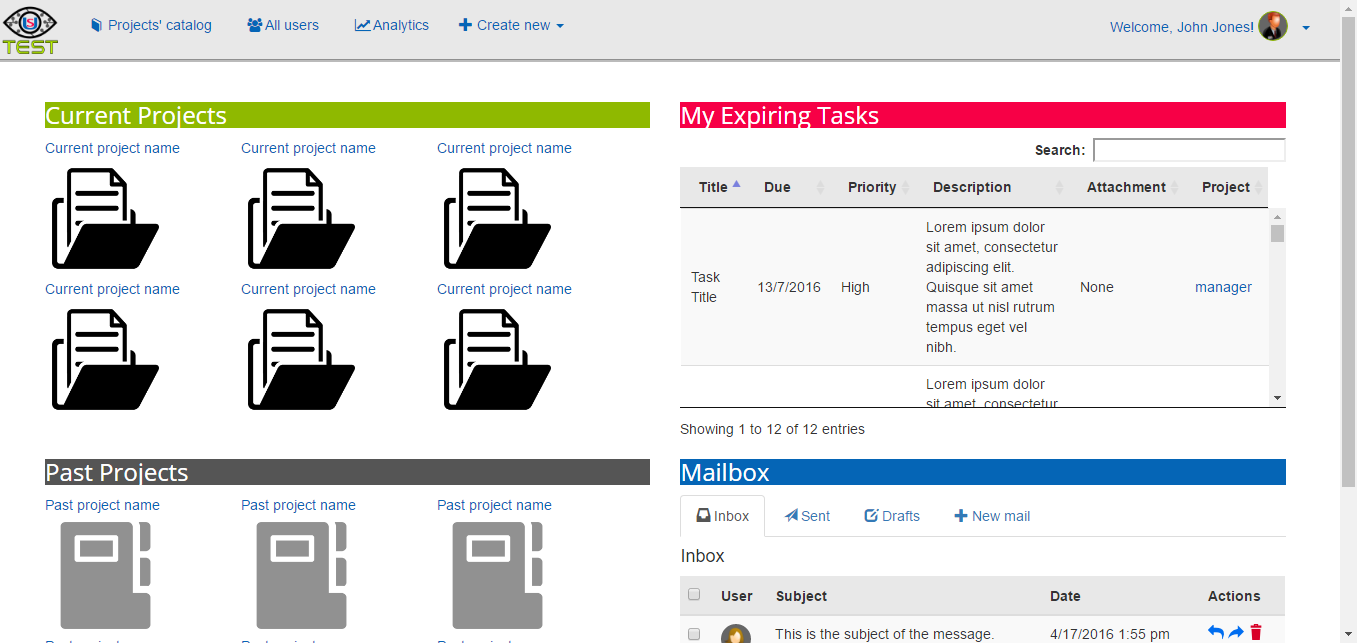
За спестяване на време и усилия при по-нататъшната разработка на прототипа е приложен еволюционния подход на прототипиране, а именно прототипът формира основите на новия продукт. Важно условие е да се използват средства, които ще бъдат прилагани при изграждане на окончателния вид на системата. Използвана е комбинация от средства за разработване на потребителски интерфейси на уеб приложения, а именно HTML, CSS, Bootstrap, jQuery, JavaScript. Прототипът се доближава до крайния вид на системата, като се характеризира с висока степен на детайлност и функционалност. Позволява на потребителите да добият представа за функционирането и визията на крайния продукт. Така се предоставя възможност в бъдеще да се извършат задълбочени тестове за ползваемост, на база на които може да се направи адекватна оценка за системата. Прототипът демонстрира стратегически важни компоненти на системата. Първоначалната реализация е на английски език, като се планира добавяне на другоезични версии на интерфейса, включително и българска.

Поради ограниченията на настоящата форма за представяне на проекта, то не е възможно да бъдат включени всички екрани. Ето защо се спираме само на някои от тях. На фиг. 3 е изобразен екранът, от който се осъществява входът в системата.



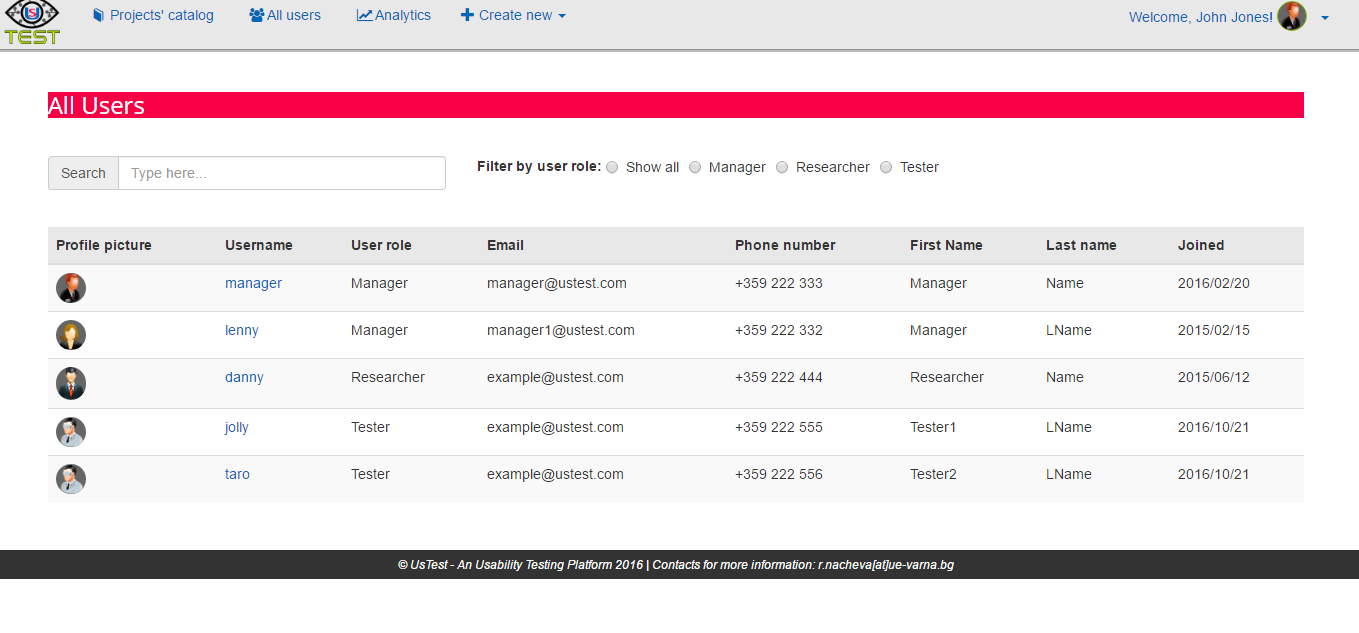
**Фиг. 3. Вход в уеб базираното приложение за управление на проекти**

На Фиг. 4 е показана стартовата страница на потребителите след успешно осъществен вход в системата.



**Фиг. 4. Начална страница на потребителите след вход**

На Фиг. 5 е изобразена страницата, на която се извежда списък с всички потребители в системата. Предоставя се възможност за филтриране на списъка по роля и ключови думи.



**Фиг. 5. Списък с всички потребители в системата**

# Заключение

В заключение може да се каже, че чрез избрания набор от технологии се цели да се постигне съвместимост между различни платформи и преносимост на кода, като се осигури стандартна работна система, чрез която може да се развие конкретния проект. Усилията са съсредоточени върху разработването на изискванията, а не върху подготовката на инструментариума за разработка на приложението. За добра организация на проекта и осигуряване на лекота при бъдещото му развиване се следват конвенции в кодирането, включително бизнес логиката е разделена от потребителския интерфейс.

Трябва да се отбележи, че прототипът не е пълно функционален, а демонстрира ключови функционалности на уеб базирана система за оценяване на ползваемостта на мобилни приложения и по-конкретно, управлението на проекти. Като следваща логична стъпка от процеса на разработване е проверката на прототипа чрез тестване с потребители. Целта е да се докаже неговата практическа приложимост и ползваемост. В тестовете е подходящо да вземат участие представители на целевата аудитория, а именно дизайнери, разработчици, мениджъри на фирми и най-вече специалисти по ползваемост.

# **Използвана литература**

1. Куюмджиев, И., Одит на информационни системи. Дис. Варна. 2011
2. Наков, Св. И др., Въведение в програмирането с Java. Фабер, Велико Търново, 2009 г.
3. Layka, V., Learn Java for Web Development. Apress, 2014
4. Williams, N., Professional: Java® for Web Applications. John Wiley & Sons, Inc., Indianapolis, Indiana, 2014

1. Cloud Native Computing Foundation. е проект на Linux Foundation, основан през 2015 г., за да подпомогне развитието на контейнерните технологии и да приведе технологичната индустрия в еволюцията си. [↑](#footnote-ref-1)
2. Куюмджиев, И. Одит на информационни системи. Дис. Варна. 2011, стр. 120 [↑](#footnote-ref-2)
3. От англ. ез. „grid system“. [↑](#footnote-ref-3)
4. Създаден е с продукта Visual Paradigm. [↑](#footnote-ref-4)