

Спецификация на изискванията на софтуера

*версия 2.0
февруари 18, 2016*

Application Store Магазин за приложения

Екип: Team Of Poxies (T.O.P.)

Елисавета Пенкова (Team leader)
Кристин Кюркчиева (Project manager)
Константин Русев (Quality manager)
Марин Воденичаров (Developer)
Стоян Денков (Q.A. тестър)

Разработва се за курса по
“Осигуряване на качеството на софтуера”
към ФМИ
на Пловдивски университет
“Паисий Хилендарси” град Пловдив

Съдържание

1. Въведение	4
1.1. Предназначение	4
1.2. Обхват на проекта	4
1.3. Речник.....	5
1.4. Препратки	5
1.5. Преглед на документа.....	6
2. Общо описание.....	7
2.1. Системна среда	7
2.2. Спецификация на функционални изисквания.....	7
2.2.1. Use-case диаграма.....	7
2.3. Потребителски характеристики.....	12
2.4. Нефункционални изисквания.....	13
3. Спецификация на изискванията.....	14
3.1. Външни изисквания към интерфейса	14
3.2. Функционални изисквания.....	17
3.2.1. Спецификация на изискванията от гледна точка на клиента	17
3.2.2. Спецификация на изискванията от гледна точка на продукта	18
3.2.3. Спецификация на изискванията от гледна точка на компонентите на продукта	19
3.3. Бъдеща работа.....	22
3.3.1. Gameplay и идея за „Flappy“	22
3.3.2. Техническа реализация.....	23
3.3.3. Потребителски интерфейс	24
Използвана литература:.....	26
Използван софтуер:.....	26

Списък с фигури

Фигура 1: Use-case диаграма.....	8
Фигура 2. Връзки в Use-case диаграмите.....	11
Фигура 3: Изглед на системата от администратор.....	14
Фигура 4.Изглед на системата от публикуващ.....	14
Фигура 5: Изглед на системата от потребител.....	15
Фигура 6: Изглед на системата от гост.....	15
Фигура 7: Списък с транзакции.....	16
Фигура 8: Списък с оферти	16
Фигура 9: База данни на приложението.....	20
Фигура 10: Връзки в базата данни.....	21
Фигура 11: Вход в играта	24
Фигура 12: Интерфейс на главното меню.....	24
Фигура 13: Интерфейс по време на игра	25
Фигура 14: Таблица с класация (leaderboard).....	25

1. Въведение

1.1. Предназначение

Целта на този документ е да представи подробно описание на система за **Application Store**. В него ще се обясни целта и функциите на системата, интерфейсите на системата, какво ще извършва системата и други. Този документ е предназначен, както за заинтересованите страни и разработчиците на системата, така и за всеки друг, който желае да продобие знание за тематиката.

1.2. Обхват на проекта

Тази софтуерна система ще бъде **Application Store** за клиент, който желае да управлява онлайн магазин за приложения. Тази система ще бъде проектирана за максимално увеличаване на производителността чрез предоставяне на средства за подпомагане на автоматизирането на процеса на преглед на приложения, качване или купуване. Чрез увеличаване на ефективността на работа и производство, системата ще отговори на потребностите на потребителите и ще бъде лесна за разбиране и използване.

По-специално, тази система е предназначена да позволи на потребителите да качват приложения, които могат да бъдат свалени безплатно или закупени от други потребители. Софтуерът ще улесни комуникацията между двата типа потребители, а цялата система ще може да бъде управлявана лесно от отделен тип потребител (администратор).

1.3. Речник

Термин	Дефиниция
Application Store	Средство (уеб страница) за продажба на приложения по електронен път.
Клиент (Client)	Лицето поръчител на този продукт (Application Store).
Гост (Guest)	Всяко лице, което има интерес към продуктите и не се е идентифицирало.
Customer	Лице, което притежава възможността да закупува приложения от магазина.
Publisher	Лице, което притежава възможността да качва нови приложения в магазина за приложения и определя цената им.
Admin	Лице, което контролира и следи всяко действие.
Спецификация на изискванията	Документ, който напълно описва всички функции на предложената система, и на ограниченията, при които трябва да работи. Например, този документ.

1.4. Препратки

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

1.5. Преглед на документа

В следващата глава от този документ се прави преглед на функционалността на продукта. Описват се неформалните изисквания и се използват за създаване на контекст за спецификацията на технически изисквания.

След това се прави спецификация предназначена основно за разработчиците, описваща подробности от техническа гледна точка за функционалността на продукта.

2. Общо описание

2.1. Системна среда

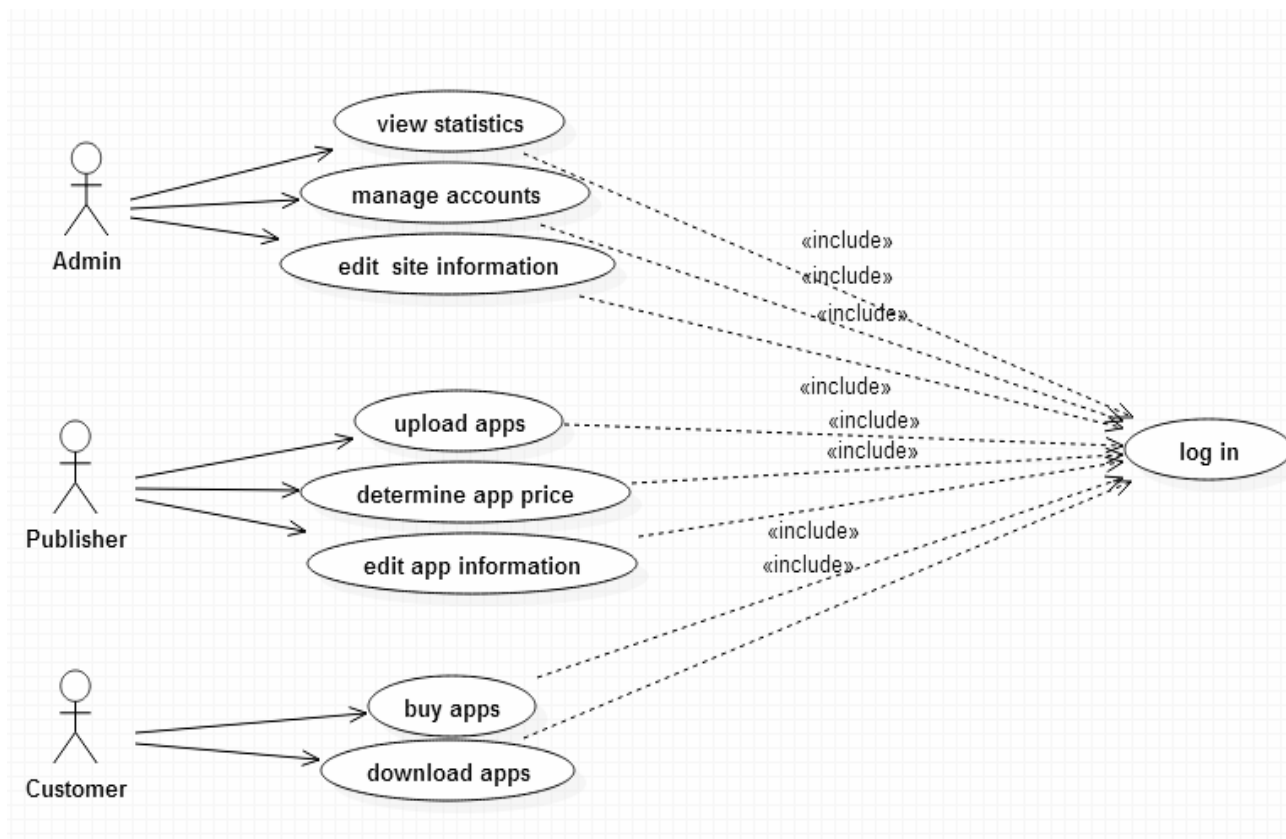
В **Application Store** съществуват 4 типа участници, гост, customer, publisher и администратор на системата. Съответно гостът има достъп само до разглеждането на продуктите и техните цени. Customer от своя страна може да преглежда и купува приложенията. Publisher има правата да регистрира приложения като безплатни или платени със съответна цена. Администраторът има достъп директно до цялата система и всички нейни функционалности.

2.2. Спецификация на функционални изисквания

В този раздел се описва използването на системата от всеки от активните участници в нея. Това са: *Administrator, Publisher, Customer*.

2.2.1. Use-case диаграма

В процеса на проектиране на софтуерния продукт, диаграмата на потребителските случаи (*use-case диаграма*) е първата диаграма, която се създава от проектантите, когато се започне проект. Тази диаграма позволява да се опишат на най-високо ниво целите на потребителя, които системата трябва да изпълнява. Тези цели не е необходимо да са задачи или действия, а може да са по общи изисквания към функционалността на системата. С други думи това е техника за определяне на функционалните изисквания на една система. Те описват типичните взаимодействия между потребителите и системата, предоставят описание на начина, по който тя се използва.



Фигура 1: Use-case диаграма

На фигурата е показана *Use-case диаграма*, описваща възможните операции на всеки един участник, имащ достъп до нашия Application Store (*Administrator, Publisher, Customer*).

Кратко описание:

Администраторът има достъп до Application Store, като има възможностите да разглежда статистика, управлява акаунтите на останалите потребители на системата и променя информацията в системата.

Преди да има достъп до цялата тази функционалност, администраторът трябва да е минал през "Log In" системата на Application Store.

Описание на use-cases (възможни услуги):

- *view statistics* - дава възможност на администраторът на системата да достъпва всякакъв вид статистика. Например: колко приложения е качил Publisher, или пък колко такива си е купил Customer и т.н.
- *manage accounts* - тази функционалност дава възможност на администраторът на системата да управлява останалите профили (акаунти) в Application Store.
- *edit the site information* - тази функционалност дава възможност на администраторът на системата да променя информацията описваща различните приложения, да я изтрива или пък добавя допълнително такава. Също така да променя, добавя и изтрива снимки към приложенията. Всичко, което може да бъде направено от останалите потребители в системата, може да бъде извършено и от администратора.

Преди да може да извършва всички тези действия, разбира се, че администраторът трябва да е преминал през "Log In" системата за идентификация.

Кратко описание:

Publisher има достъп до Application Store, като има възможностите да качва в системата нови приложения, да добавя съответна цена за тях, както и да променя информацията на качените от него приложения. Преди да има достъп до цялата тази функционалност, трябва да е преминал през "Log In" системата на Application Store.

Описание на use-cases (възможни услуги):

- *upload apps* - тази функционалност дава възможност на даден publisher да добавя нови приложения или игри към магазина за приложения.
- *determine app price* - тази функционалност дава възможност на publisher сам да определи и постави цената на своя продукт (приложение).
- *edit app information* - тази функционалност дава възможност на даден publisher да променя, добавя и изтрива текстовата или снимкова информация за своето приложение.

Преди да може да извършва всички тези действия, разбира се, че публикуващият трябва да е преминал през "Log In" системата за идентификация.

Кратко описание:

Всеки **Customer** има достъп до Application Store, като притежава възможностите да купува приложения или пък свободно да изтегля такива, ако те нямат поставена цена или пък цената им е нулева. Също така има възможност за микротранзакции. Например, дадено приложение може да бъде свободно за сваляне, но в него да има вградени функции, които да бъдат платени. Освен това обикновеният потребител има възможност да разглежда своите постижения (achievements) или да види класацията на всички потребители използвали дадено приложение (leaderboard).

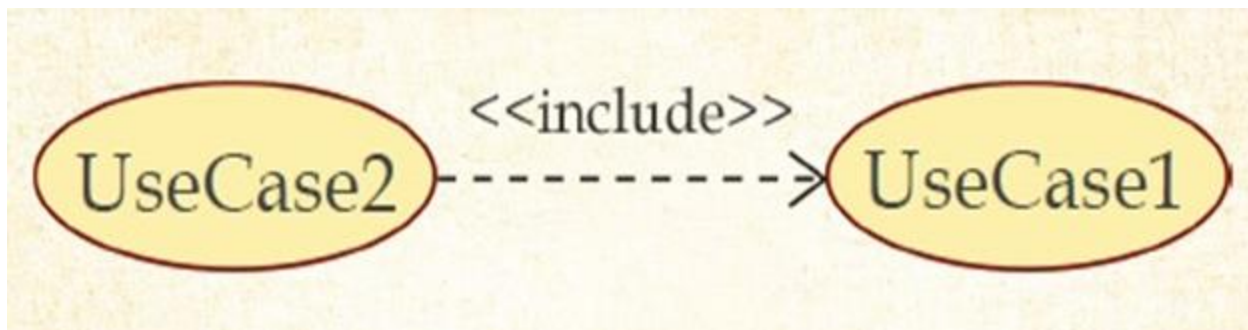
Описание на use-cases (възможни услуги):

- *buy apps* - тази функционалност дава възможност на потребителя (customer) да купува избрано от него приложение и след това да може да го изтегли свободно. Цената е поставена от автора на приложението.
- *download apps* - тази функционалност дава възможност на потребителя (customer) да изтегля избрано от него приложение свободно. Авторът на приложението е решил да го качи за свободно изтегляне или пък с нулева цена. След като веднъж е купено приложението, то може да бъде изтеглено произволен брой пъти на различни устройства.

Преди да може да извършва всички тези действия, разбира се, че потребителят (customer) трябва да е преминал през "Log In" системата за идентификация.

Връзки в use-case диаграмите:

- *Directed association* - актьорът стартира use-case, тоест казва на системата да извърши услугата, записана в use-case.
- *Dependency между use cases* - Разширяване на функционалността.
- *Include (uses)* - Първо се изпълнява use-case, към който е насочена връзката и след това се изпълнява use case, от който идва връзката. Например: *UseCase1* се изпълнява, после се изпълнява *UseCase2*:



Фигура 2. Връзки в Use-case диаграмите

Когато клиентът разгледа изготвените Use-case диаграми, вижда как ще може да използва системата и дали услугите, които предоставя тя, покриват неговите изисквания. В този момент още, преди да е започнало разработването на софтуера, той има възможност да приеме предложената функционалност или да обсъди с анализатора нейната промяна.

2.3. Потребителски характеристики

Всеки Customer се очаква да притежава добра компютърна грамотност и да може да използва Application Store и неговите функционалности.

Основната страница на **Application Store** ще предоставя възможност за разглеждане на всички приложения или игри в онлайн магазина.

Публикуващият приложения (Publisher) се очаква да бъде достатъчно добре подготвен и компютърно грамотен, за да може да използва Application Store като средство да продава своите продукти.

Администраторът се очаква да бъде достатъчно добре запознат със системата и нейните функционалности за да може успешно да управлява онлайн магазина за приложения.

Подробна информация и снимков материал за външния вид на Application Store ще се разгледа в раздел три по-долу в документа.

2.4. Нефункционални изисквания

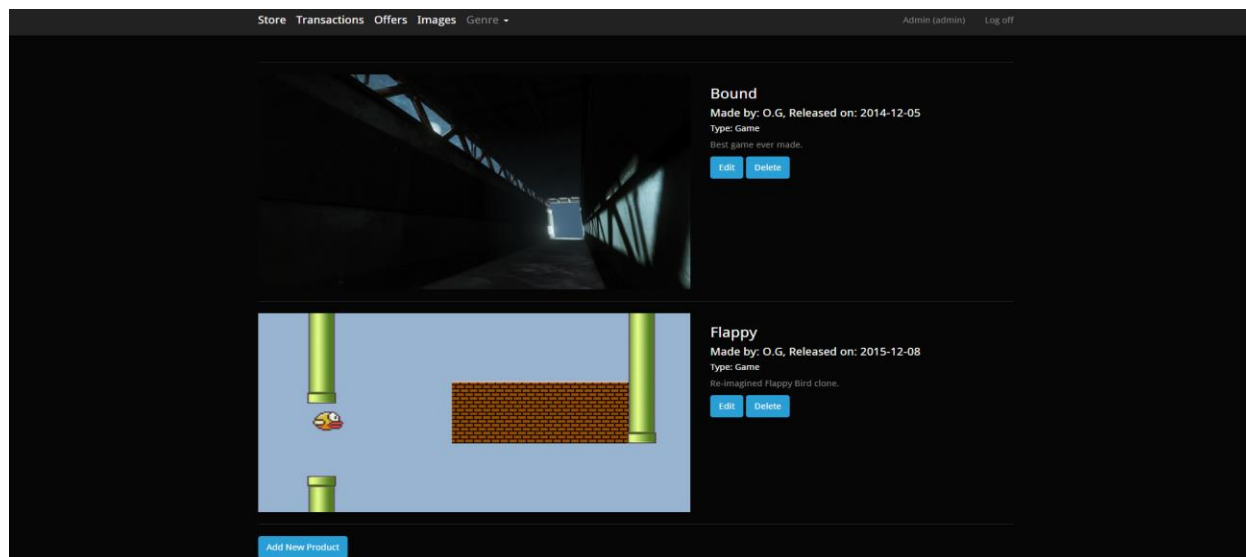
Application Store ще бъде на сървър с високоскоростни интернет възможности. Софтуерът, разработен тук предполага използването на инструмент като ASP.NET MVC за връзка между уеб страниците и базата данни. Скоростта на връзката ще зависи от използвания хардуер, повече отколкото характеристиките на тази система.

ASP.NET MVC е платформа, създадена от Microsoft, която служи за изработване на уеб приложения, използвайки модела Model-View-Controller (MVC). Платформата използва C#, HTML, CSS, JavaScript и бази данни. ASP.NET MVC е съвременно средство за изграждане на уеб приложения, което не замества изцяло уеб формите. Платформата включва нови тенденции в разработката на уеб приложения, притежава много добър контрол върху HTML и дава възможност за създаване на всякакви приложения. ASP.NET MVC може да бъде много лесно тествана и допълвана, защото е изградена от отделни модули, които са изцяло независими едни от други. Чрез платформата се създават цялостни приложения, които се стартират, а не единични скриптове.

3. Спецификация на изискванията

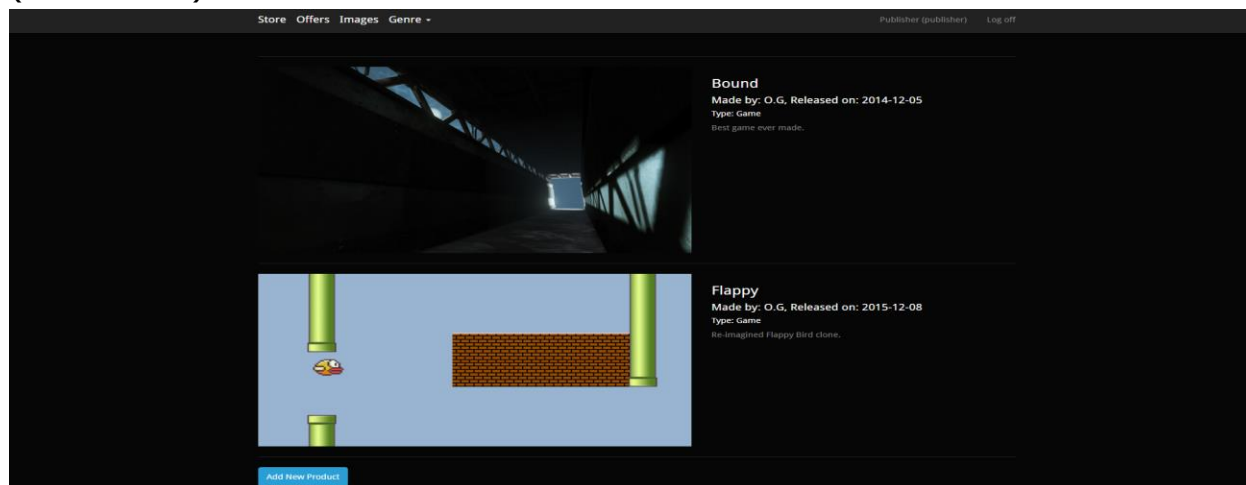
3.1. Външни изисквания към интерфейса

Прилагаме фигури описващи и показващи интерфейса на Application Store. На фигурата по-долу се показва изгледът на онлайн магазина за приложения от гледна точка на администраторът на системата.



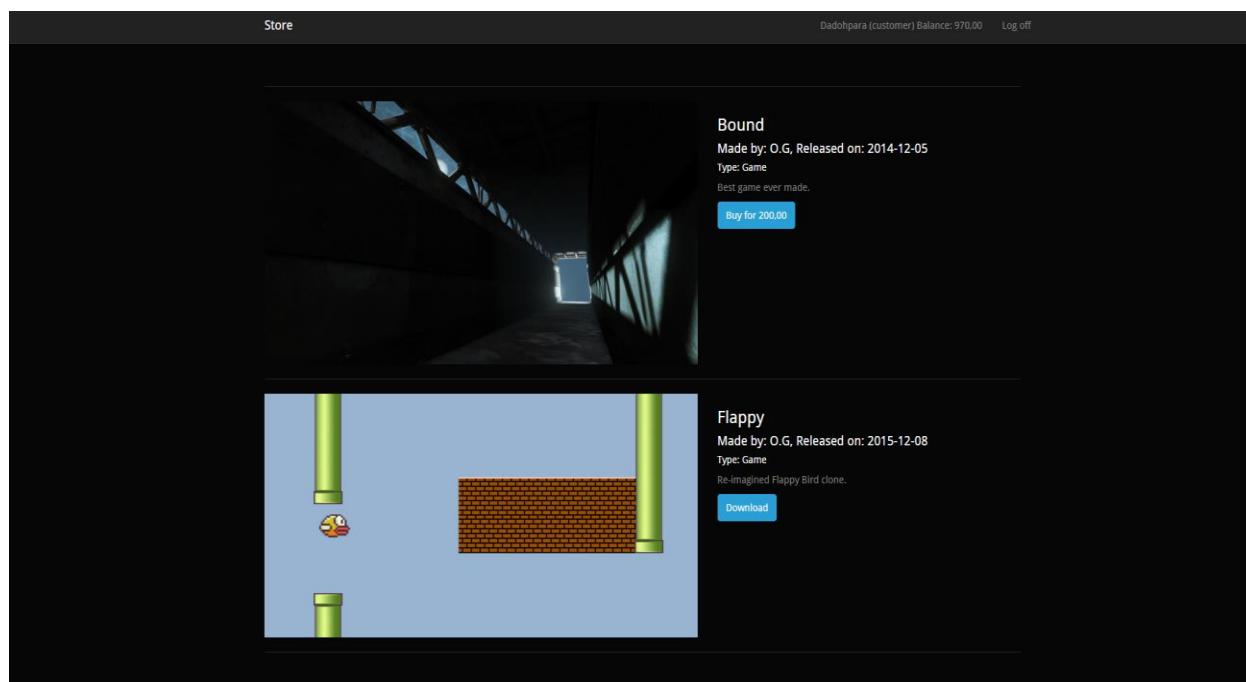
Фигура 3: Изглед на системата от администратор

На следващата фигура се показва изгледът на онлайн магазина за приложения от гледна точка на потребителите публикуващи приложения (Publishers).



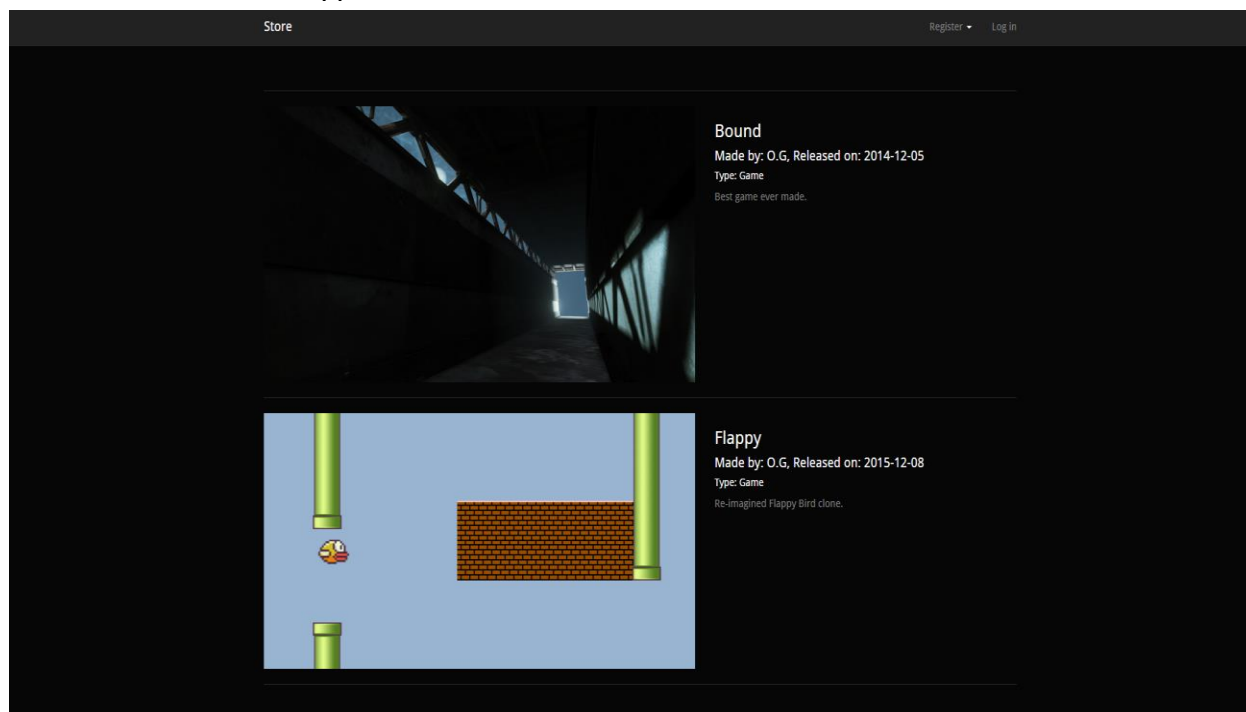
Фигура 4.Изглед на системата от публикуващ

От гледна точка на Customers, системата изглежда по следния начин:



Фигура 5: Изглед на системата от потребител

Изгледът за потребители (гости), които не са преминали през Log In системата е следният:



Фигура 6: Изглед на системата от гост

Store	Transactions	Offers	Images	Genre ▾	Admin (admin)	Log off
-------	--------------	--------	--------	---------	---------------	---------

ApplicationUserID	UserName	Title	OfferType	Description	Amount	Date
14d889b9-e9f5-4257-9ca0-dc35399c15ab	Customer	Flappy	Purchase	Game purchase	0,00 лв.	2015-12-09
14d889b9-e9f5-4257-9ca0-dc35399c15ab	Customer	Bound	Purchase	Game purchase	200,00 лв.	2015-12-09
14d889b9-e9f5-4257-9ca0-dc35399c15ab	Customer	Flappy	IAP	Enable music	0,69 лв.	2015-12-09
14d889b9-e9f5-4257-9ca0-dc35399c15ab	Customer	Flappy	IAP	Flappy chrome	5,92 лв.	2015-12-09
14d889b9-e9f5-4257-9ca0-dc35399c15ab	Customer	Flappy	IAP	Ram bricks	2,16 лв.	2015-12-09
14d889b9-e9f5-4257-9ca0-dc35399c15ab	Customer	Flappy	IAP	SAVAGE	13,37 лв.	2015-12-09
14d889b9-e9f5-4257-9ca0-dc35399c15ab	Customer	Flappy	IAP	Gray up-pipe	0,73 лв.	2015-12-09
14d889b9-e9f5-4257-9ca0-dc35399c15ab	Customer	Flappy	IAP	Gray down-pipe	0,73 лв.	2015-12-09
14d889b9-e9f5-4257-9ca0-dc35399c15ab	Customer	Flappy	IAP	Maroon background	2,78 лв.	2015-12-09
14d889b9-e9f5-4257-9ca0-dc35399c15ab	Customer	Flappy	IAP	Lavender background	3,14 лв.	2015-12-09
14d889b9-e9f5-4257-9ca0-dc35399c15ab	Customer	Flappy	IAP	Ivory background	2,19 лв.	2015-12-09
14d889b9-e9f5-4257-9ca0-dc35399c15ab	Customer	Flappy	IAP	LightGreen background	1,11 лв.	2015-12-09

Фигура 7: Списък с транзакции

Store	Transactions	Offers	Images	Genre ▾	Admin (admin)	Log off
-------	--------------	--------	--------	---------	---------------	---------

ID	Title	Description	OfferType	Price	
1	Bound	Game purchase	Purchase	200,00 лв.	Edit Delete
2	Flappy	Game purchase	Purchase	0,00 лв.	Edit Delete
3	Flappy	Enable music	IAP	0,69 лв.	Edit Delete
4	Flappy	Flappy chrome	IAP	5,92 лв.	Edit Delete
5	Flappy	Ram bricks	IAP	2,16 лв.	Edit Delete
6	Flappy	SAVAGE	IAP	13,37 лв.	Edit Delete
7	Flappy	Gray up-pipe	IAP	0,73 лв.	Edit Delete
8	Flappy	Gray down-pipe	IAP	0,73 лв.	Edit Delete
9	Flappy	Maroon background	IAP	2,78 лв.	Edit Delete
10	Flappy	Lavender background	IAP	3,14 лв.	Edit Delete
11	Flappy	Ivory background	IAP	2,19 лв.	Edit Delete
12	Flappy	LightGreen background	IAP	1,11 лв.	Edit Delete

[Add New Offer](#)

Фигура 8: Списък с оферти

3.2. Функционални изисквания

Крайният потребител на разработвания **Application Store** купува или публикува приложения. Publisher (или съответно организацията, която този publisher представлява) осигурява приложения в реално време, а Customer може да ги придобие чрез покупко-продажба (веднъж закупено приложение, може да бъде свободно придобито от неговия собственик на други устройства).

3.2.1. Спецификация на изискванията от гледна точка на клиента

Клиент, имащ желание да управлява електронен магазин за приложения (Application Store), заявява своите претенции относно функционалността. Той желае в магазина да има три типа участници:

- Хора, които могат да публикуват приложения;
- Потребители, които да купуват тези приложения;
- Администратор, който да следи (и управлява) цялата дейност, извършвана в магазина.

Клиентът желае този продукт да бъде реализиран като уеб сайт, който да може да бъде достъпван както от компютър, така и от телефон или таблет.

Освен да качва приложения, потребителят, който публикува приложения (publisher), трябва да може да определя цената и да добавя описание в текстов и снимков формат.

Клиентът няма претенции относно използваните техники за реализацията на онлайн магазина за приложения. Единственото, условие е да получи краен продукт, имащ желаната функционалност със семпъл, но стилизиран дизайн.

3.2.2. Спецификация на изискванията от гледна точка на продукта

Целта е да се направи дизайн и прототипна имплементация на сървър за приложения и примерни приложения със следните характеристики:

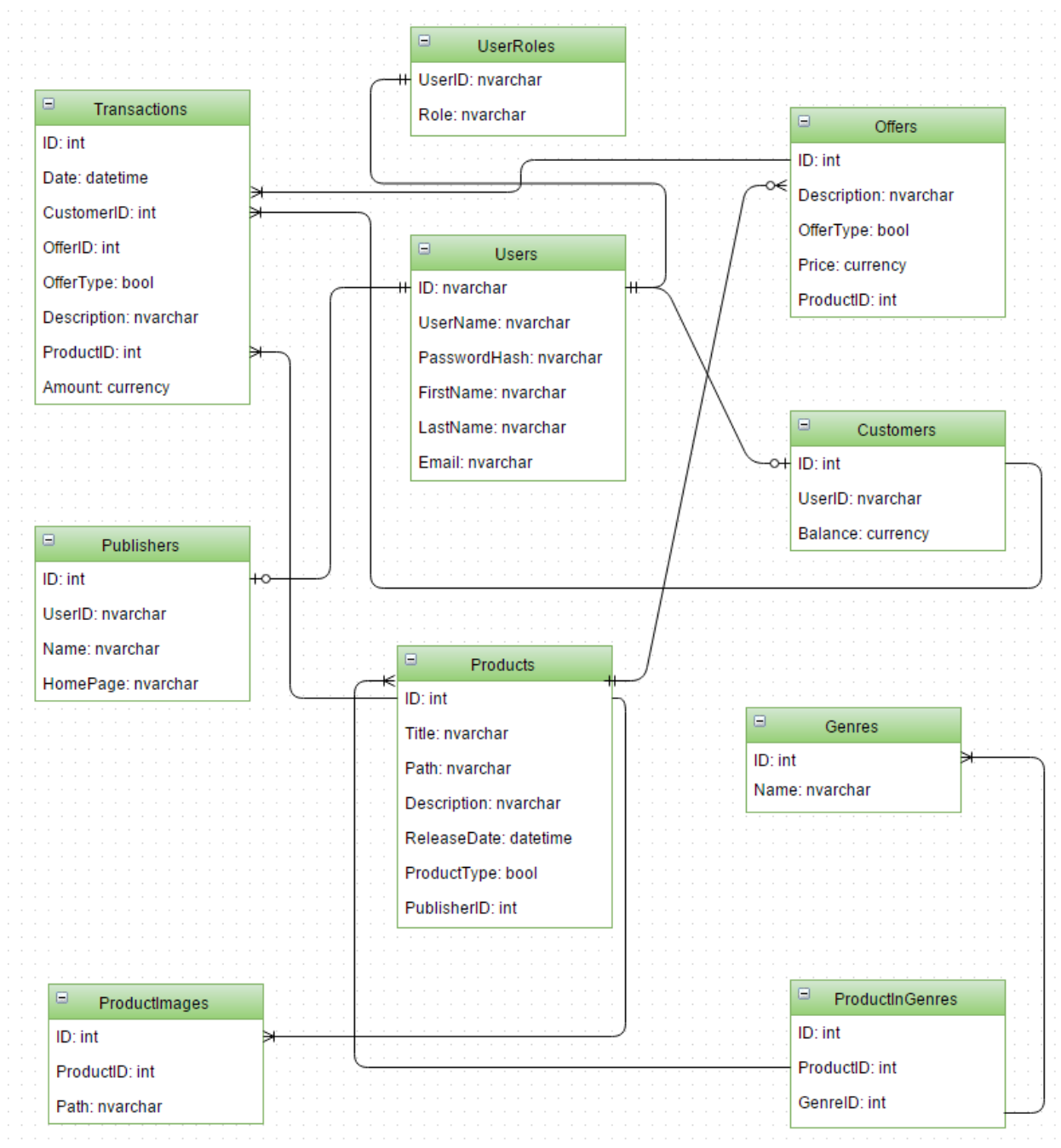
Сървър за приложения/Магазин:

- Потребителска аутентикация. Приложенията ползват потребителски акаунт за достъп до АПИ-то на сървъра.
- Потребителите могат да регистрират приложения в сървъра като безплатни или платени със съответна цена.
- Потребителите могат да преглеждат и купуват приложенията регистрирани на сървъра.
- При отваряне на приложения, сървърът генерира анонимен потребителски идентификатор (UserID) и го подава на приложението.
- Приложенията, посредством АПИ-то на сървъра, могат да проверят дали даден потребител (UserID) е закупил / има достъп / до тях.
- Приложенията, посредством АПИ-то на сървъра, могат да подават заявки за осъществяване на микро-транзакции за определена сума от определен потребител (UserID). Да съществува и механизъм за проверка за успешно осъществяване на транзакцията.
- Да се пази списък на всички закупени приложения и осъществени микро-транзакции от потребителите.

Крайният проект трябва да може да бъде използван от три отделни платформи: сървър за приложения; сървър на конкретно приложение; клиент (десктоп, уеб или мобилен).

3.2.3. Спецификация на изискванията от гледна точка на компонентите на продукта

- Архитектурен дизайн: трябва да бъде реализирана коректна комуникация между всеки два от трите вида участници в архитектурата (клиент-сървър, клиент-приложение, приложение-сървър).
- Инфраструктурен дизайн: трябва да бъдат използвани подходящи езици, технологии и работни рамки за улесняване бъдещото доразвиване на проекта.
- Техническо изпълнение: отделните части на проекта трябва да са добре написани, не зависят силно една от друга и работят коректно.
- API (Application Programming Interface): В разработвания проект приложно-програмният интерфейс се използва за комуникация между сървъра на Application Store и сървърите на публикуваните от Publishers приложения. Като целта на API-то е да позволи на клиентите да правят транзакции директно през приложението, което използват. Например извършването на плащане за допълнителна функционалност. Всеки от разработчиците на публикуваните в магазина приложения (които приложения са платени или съдържат микро-транзакции) би следвало също да реализират свое API, комуникиращо с това. За да може да се случи това е нужно да се създаде допълнителна документация, за да могат да знаят разработчиците до каква функционалност имат достъп и как да я използват.
- База Данни: илюстрация на базата данни в контекста на Entity-relationship диаграмите. Въпреки че, ER предлага възможност за изобразяване на many-to-many връзки, на диаграмата е използвана помощна таблица, където е нужна такава връзка, като целта е намалянето на абстракцията между диаграмата и реализацията в езика за програмиране.



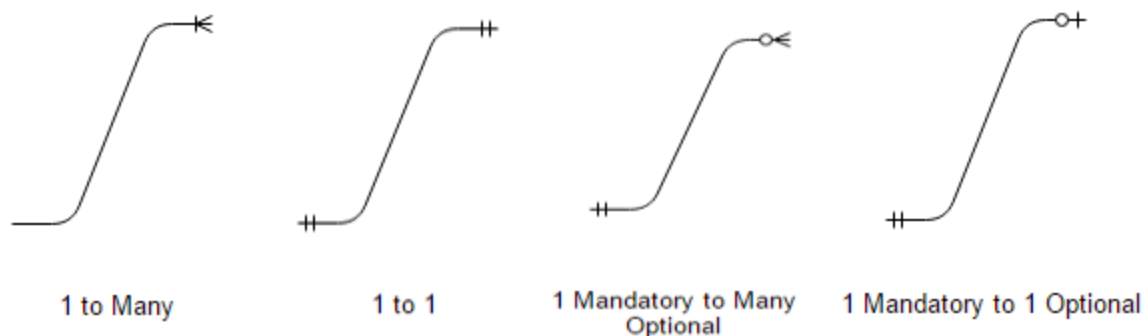
Фигура 9: База данни на приложението

Някои нужни пояснения са булевите стойности на полетата OfferType и ProductType. Целта на OfferType е да индикира дали офертата е еднократна покупка, т.е. за достъп до свалянето на продукта или е микротранзакция, извършваща се след като потребителя има достъп до вече закупения от него продукт. ProductType служи за разграничаването на игрите от приложенията.

Полето Role в таблица UserRoles е предназначено за вече споменатите роли - customer, publisher и admin.

Друго специфично нещо е дублирането на информация между таблиците с офертите и тази с транзакциите. Причината за това е да подпомогне запазването на информацията за извършените транзакции, тъй като оферите подлежат на промяна и изтриване от публикуващите (publishers).

С цел да се избегнат неяснотите, по-долу са приложени използваните връзки.



Фигура 10: Връзки в базата данни

3.3. Бъдеща работа

Да бъде реализирана игра „Flappy“, която да служи за тестова игра в Application Store. Целта е да се проследи дали всички транзакции минават безпроблемно, да бъде доуточнено къде се съхраняват постиженията (achievements) и leaderboards.

Най-важният аспект е дали създаденото API е достатъчно функционално и скалируемо с нуждите на бъдещите Publisher-и. Също така „Flappy“ и неговото API ще бъдат пример за това как трябва разработчиците, които ще продават в магазина, да реализират своите API-та за комуникация с магазина.

3.3.1. *Gameplay* и идея за „Flappy“

Играта е базирана на популярната игра „Flappy Bird“, но е преработена така, че да има разнообразен вид препятствия и да има интересни за играчите микротранзакции, achievements, както и leaderboards. Играта представлява *side-scroller* и целта е навигирането на една птица през препятствия – тръби и тухли. Управлението на птицата е измамно просто – натискането на един клавиш, което да представлява изкачване. Измамното е, че има много малък интервал от време, в който птицата трябва да се изкачи или се разбива и също така при появата на тухли, не винаги е ясно от къде е изходът, което сериозно увеличава комплексността, тъй като препятствията се генерират на произволен принцип (но така, че винаги има момент, в който преминаването да е възможно).

3.3.2. Техническа реализация

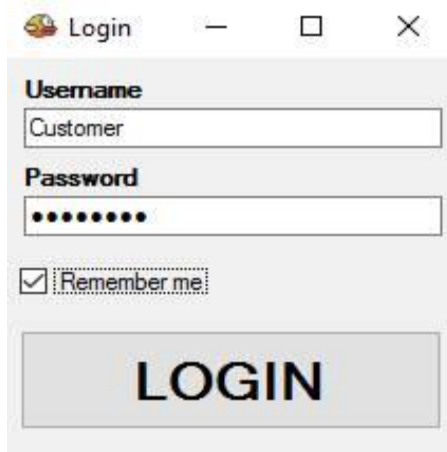
Първо трябва да бъде създаден сървър за играта - FlappyServer. Целта му ще бъде да проверява самоличността на потребителя и да запазва точките му за класациите (leaderboards), както и неговите постижения (achievements).

Следва реализацията на самата игра. За целта ще се използват Windows Forms и C#. Предимствата на този избор са, че API-то (на играта, тоест различно от това на FlappyServer) ще се реализира бързо и лесно. Недостатъци са кадрите в секунда, които ще бъдат генерирани и всички останали ограничения породени от използването на Windows Forms. Основната техника, която ще се използва е създаването на времеви ограничители (timers) и прерисуването на формата в определен момент от време, което всъщност ще представлява движението на обектите по екрана. Разбира се самата птица няма да се движи, а препятствията ще се движат към нея, което ще създаде достатъчно добра илюзия за движение.

Нужно е да се създаде и Login Form, която да използва регистрацията от **Application Store**. По този начин може да се идентифицира потребителят и да му бъдат асоциирани точки и постижения, както и да бъдат осъществявани транзакции по негово желание. Някои примерни микротранзакции са например - визуални промени върху птицата и препятствията, смяна на музиката или промяната в трудността на играта.

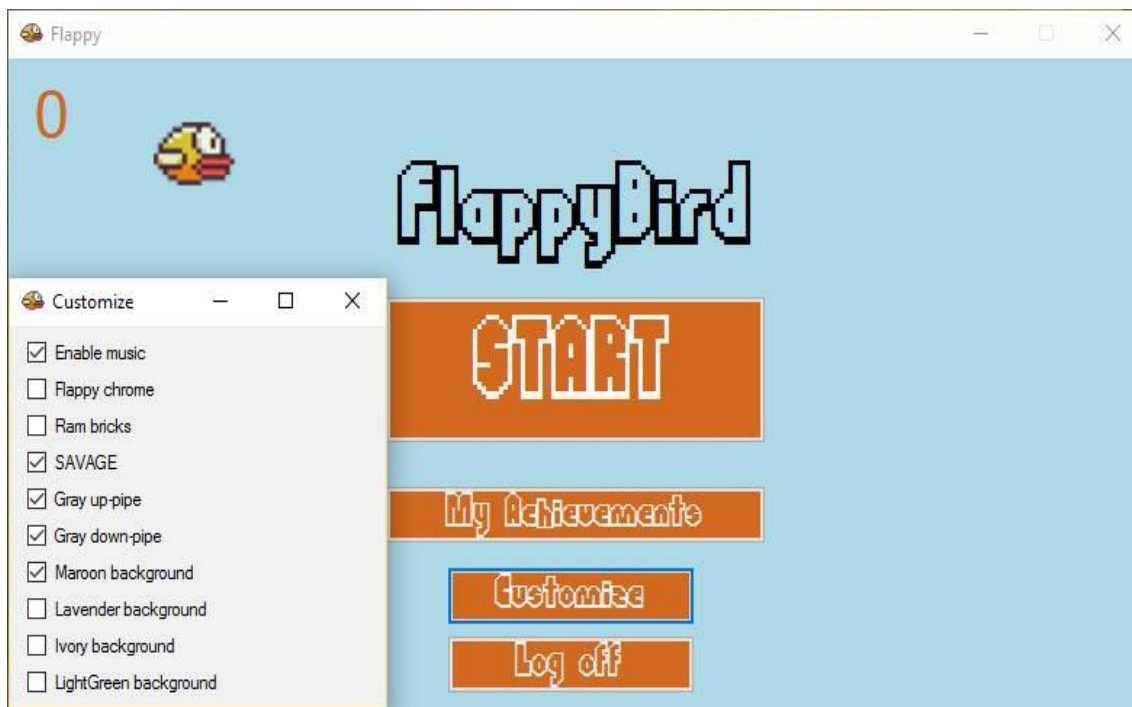
3.3.3. Потребителски интерфейс

След успешен логин (вход в играта) потребителят влиза в менюто, където може да преглежда постиженията си.

A screenshot of a 'Login' window. It has a title bar with a small bird icon and the text 'Login'. Inside, there are two text input fields: 'Username' with the text 'Customer' and 'Password' with masked characters '.....'. Below these is a checkbox labeled 'Remember me' which is checked. At the bottom is a large button labeled 'LOGIN'.

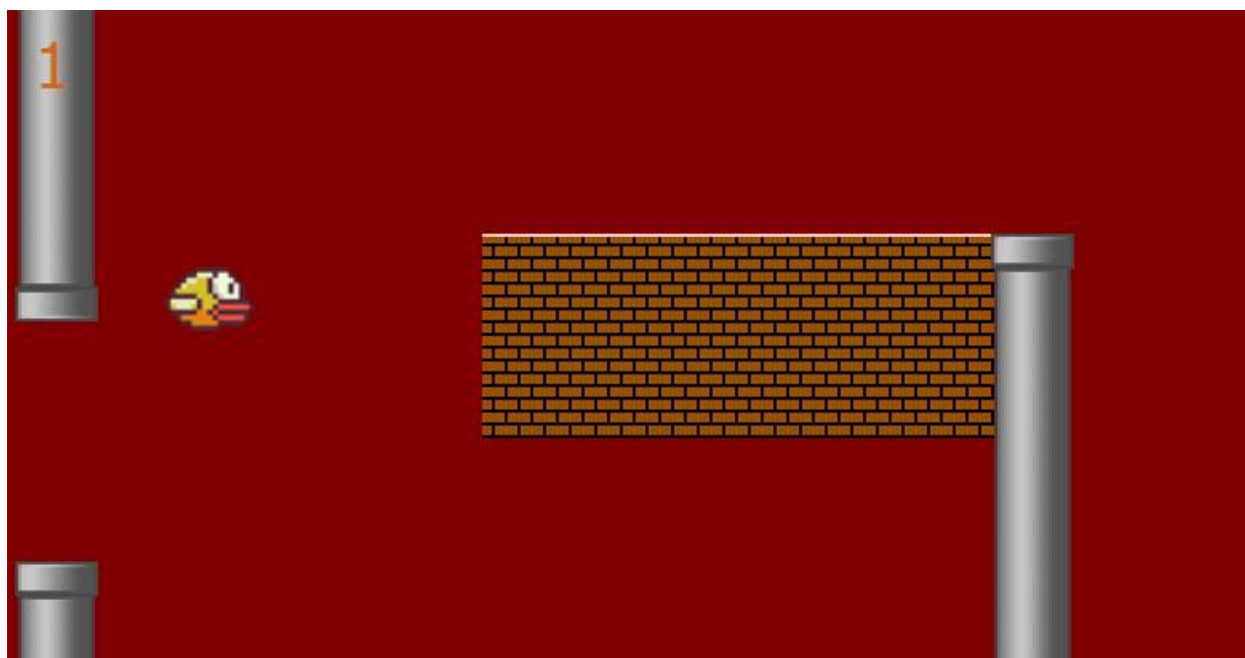
Фигура 11: Вход в играта

Също така може да преглежда и какво има за закупуване или да избира какво от закупеното да използва в следващата игра.



Фигура 12: Интерфейс на главното меню

При стартиране на играта в горния ляв ъгъл има брояч на точките, тоест броят преминати препятствия.



Фигура 13: Интерфейс по време на игра

След като птицата не успее да премине през препятствие или падне, тоест се разбие, се показва таблицата с класациите (leaderboard).

UserName	Score
Customer	19
Customer	8
Customer	4
Customer	3

Фигура 14: Таблица с класация (leaderboard)

Използвана литература:

- „Ръководство по софтуерни технологии“, Пловдивски университетско издание 2003 година (Станимир Стоянов, Ася Стоянова-Дойчева, Малина Трендафилова, Емил Дойчев)
- Лекции от курса по софтуерни технологии за специалност информатика при ФМИ на Пловдивски университет „Паисий Хилендарски“ гр. Пловдив
- Лекции от курса по осигуряване на качеството на софтуера (Q.A.) за специалност информатика при ФМИ на Пловдивски университет „Паисий Хилендарски“ гр. Пловдив
- Примери от упражнения по софтуерни технологии и осигуряване на качеството на софтуера
- Уикипедия
- ESI Center Eastern Europe

Използван софтуер:

- *StarUML*: Софтуер за изготвяне на UML диаграми.
- *Microsoft Office Packet 2013*: За изготвяне структурата на документа и представяне.
- *Google Drive*: За едновременна работа между членовете на екипа.
- *Adobe Reader*: За представяне на документа.
- Онлайн софтуер за разработка на ER диаграми