

Курсов проект

Дисциплина: Проектиране и интегриране  
на софтуерни системи

Фаза 2: Анализ на изискванията и  
проектиране на системата

*Student's newsletter.*

Фак. №	Име на студент	Секция от документа
61911	Денис Дуев	Въведение, системна архитектура
61893	Александър Илиев	Анализ на изискванията, Проектиране на потребителски интерфейс
61958	Мартин Йотов	Обобщение на потребитския интерфейс
61879	Михаела Чакова	Матрица на изискванията, Activity diagrams

Ноември, 2018

# Съдържание

<b>1</b>	<b>ВЪВЕДЕНИЕ</b>	<b>3</b>
1.1	Цел	3
1.2	Обхват	3
1.3	Резюме	4
1.4	Дефиниции и акроними	4
<b>2</b>	<b>АНАЛИЗ НА ИЗИСКВАНИЯТА</b>	<b>5</b>
2.1	Функционални изисквания	5
2.2	Нефункционални изисквания	7
2.2.1	Изисквания към потребителския интерфейс	7
2.2.2	Производителност	7
2.2.3	Наличност	7
2.2.4	Сигурност	7
<b>3</b>	<b>СИСТЕМНА АРХИТЕКТУРА</b>	<b>8</b>
3.1	Архитектурно проектиране	8
3.2	Описание на декомпозицията	9
3.3	Обосновка на проектирането	10
<b>4</b>	<b>ДАННОВО ПРОЕКТИРАНЕ</b>	<b>11</b>
4.1	Описание на данните	11
4.2	Речник на данните	11
<b>5</b>	<b>ПРОЕКТИРАНЕ НА ПОТРЕБИТЕЛСКИЯ ИНТЕРФЕЙС</b>	<b>11</b>
5.1	Обобщение на потребителския интерфейс	11
5.2	Екранни изображения	14
<b>6</b>	<b>МАТРИЦА НА ИЗИСКВАНИЯТА</b>	<b>17</b>

# 1 ВЪВЕДЕНИЕ

## 1.1 Цел

Целта на този проект е да се опишат изискванията за софтуерната система, анализа на тези изисквания и да се опише проектирането на системата. Основните неща които ще разгледаме в този документ са свързани още с архитектурата на самата система, на отделните компоненти и обосновка на избраната архитектура. Документа още включва проектирането на данните, както и какво данни ще се използват в системата, как ще бъдат предавани между различните агенти, как ще бъдат съхранява и къде. Ще разгледаме още различните компоненти и как те са проектирани. След това може да видите кратко обяснение за потребителския интерфейс както и негово първоначално проектиране. Накрая ще видите и матрицата на изискванията.

## 1.2 Обхват

Софтуерната система цели да обхване студентите от даден университет. Често новините свързани с университета не достигат до всички студенти. Използват се различни канали за споделяне на информация и често покрай това има неразбирателства.

Чрез системата Student's newsletter ще предоставим web интерфейс, така че всички студенти от даден университет, които имат нужда да виждат новини да общуват и обсъждат нещата в университета. Системата може да се разглежда от две перспективи.

Първата е че всеки университет ще си има собствена секция за новини и чат канал. Съответно цялата система може да бъде разработена като сбор от много по-малки системи. По този начин ще се разпредели потока от данни, съответно всеки университет сам ще може да поддържа своята част от системата.

Втората перспектива е системата като, тя ще предоставя възможността потребителя да избере своя университет и съответно да достъпва само конкретната система. На пръв поглед системата ще изглежда като едно, но ще бъде съставена от множество по-малки системи за различните университети, които ще предоставят обработката и предоставянето на данни.

По този начин нови университети лесно могат да се интегрират в цялостната система и тя да се развива.

Целта е да се предостави прозрачност на нещата ставащи в университета и с неговото участие, така студентите ще се чувстват по-близки до средата и по-ангажирани. По този начин използвайки единен интерфейс и множество под системи, много лесно ще се постигне еднакво потребителско изживяване. Идеята, че студенти от различни университети ще могат да ползват на пръв поглед една и съща система, но информацията представена таб ще бъде предоставена от под системата на конкретния университет.

## 1.3 Резюме

Документът описва началната фаза на софтуерния проект, като включва предварителните фази по определяне и анализиране на изискванията. Първата част която ще видите е именно въведението съдържащо общата информация за софтуерния проект, като целта и обхвата. Идеята е накратко да се запознаете с идеята зад софтуерния проект. Също така в тази първоначална секция можете да намерите и частта с дефиниции и акроними. Там ще можете да проверите всички понятия които се използват в този документ както и специфичните акроними свързани с проектиране на софтуерната система. Следващата част от документа която ще разгледаме е анализа на изискванията. След като сме събрали изисквания за системата от различните стейкхолдъри. Те са разделени на две основни категории. Функционалните изисквания и нефункционалните изисквания, които включват в себе изискванията към потребителския интерфейс, към производителността, изискванията към наличността и сигурността, както и дали системата ще отговаря на конкретните стандарти. Следва архитектурата на системата, там в различни секции са разгледани архитектурното проектиране, описание на декомпозицията на системата и обосновка защо системата е проектирана по този начин. Следващата главна секция съдържа в себе си описание на данното проектиране, а именно описание на данните и речник на данните. После може да намерите компонентното проектиране, което да съдържа основните главни компоненти на системата, приложени стратегии и алгоритми.

Потребителският интерфейс е следващата главна част. Там ще може да видите кратко обобщение на потребителския интерфейс, както и евентуални скрийншоти от мокъпи на потребителския интерфейс. Накрая ще намерите матрицата на изискванията - представена под формата на таблица и последната секция с приложения, където ще намерите всякаква друга допълнителна информация за софтуерната система, която по не е намерила място в останалите секции..

## 1.4 Дефиниции и акроними

Стейкхолдър - заинтересовано лице за софтуерната система. Лице, което е пряко свързано с разработката, поддръжката и използването на софтуерната система.

мокъп - примерно представяне на потребителския интерфейс, имплементиран без съответната му програмна логика, целящ да представи последователността от действия които потребителя може да избере, както и примерно представяне/подреждане на елементите.

Subscribe - записване за получаване на конкретна новина, абониране. В случая потребителя се абонира да получава съобщения от останалите потребители, чрез чат сървър.

*Дефинирайте всички термини, понятия и акроними, използвани в документа.*

## 2 АНАЛИЗ НА ИЗИСКВАНИЯТА

Изискванията са приоритизирани по следните критерии:

- Приоритет 1 - “задължително” изискване
- Приоритет 2 - “необходимо” изискване
- Приоритет 3 - “препоръчително” изискване
- Приоритет 4 - “маловажно” изискване

### 2.1 Функционални изисквания

Идентификатор	Изискване	Коментар	Приоритет
FR_01	Системата задължително трябва да предоставя начин за регистрация на потребители	За регистрация трябва да се използва университетския e-mail и имената на потребителя	1
FR_02	Системата задължително трябва да предоставя начин за вход с e-mail и парола	Регистрираните вече потребители, трябва да могат да влязат, използвайки e-mail адреса с който са се регистрирали	1
FR_03	Необходимо е системата да показва информация само на регистрирани потребители	Новините и чатове да са недостъпни без вход в системата	2
FR_04	Системата задължително да позволява качването на новини	Администраторите да могат да създават новини(съобщения)	1

<b>FR_05</b>	Системата задължително да позволява презентирането на новини	След като дадена новина е създадена, да може да се избира дали да бъде видима и ако е да бъде достъпна за всички потребители	1
<b>FR_06</b>	Системата задължително поддържа чат между потребителски групи	Възможност за изпращане и получаване на съобщения в дадена група от всеки тип потребители	1
<b>FR_07</b>	Системата е необходимо да има една обща чат група	Съществува общ чат, в който всеки потребител участва	2
<b>FR_08</b>	Препоръчително е системата да има отделни чат група за всяка новина	Потребителите могат да се включат в чат група за всяка публикувана новина	3
<b>FR_09</b>	Задължително е потребителите да могат да сменят паролата си за достъп	Да има начин, по който потребителят може да смени паролата си	1
<b>FR_10</b>	Необходимо е да съществува начин за промяна на личната информация за потребителя (Имена)	Редакция на имената въведени от потребителя при регистрация (Тези имена са видими за всеки потребител, който използва системата)	2
<b>FR_11</b>	Препоръчително е да има някакъв тип известяване при наличие на ново съобщение	При наличие на ново съобщение в група, в която потребителя участва, да бъде уведомен по някакъв начин	3
<b>FR_12</b>	Необходимо е списъкът с новините да може да бъде	Да съществува бутон или движение при което се проверява за нова	2

	обновяван при желание от потребителят по всяко време	информация и се показва в списъкът с новини	
<b>FR_13</b>	Маловажно изискване е системата да има светла и тъмна тема	Потребителят да може да избира начина по който вижда системата измежду две теми - светла и тъмна	4
<b>FR_14</b>	Необходимо е да съществува начин да се уведомят всички потребители за важна новина	Администраторите имат възможност да уведомят всички потребители за публикуването на важна новина	2

## 2.2 Нефункционални изисквания

### 2.2.1 Изисквания към потребителския интерфейс

- Визуализацията да следва Material Design
- Системата няма да поддържа обратна връзка чрез звук

### 2.2.2 Производителност

- Системата да е в състояние да понесе трафик до 20 000 души без това да оказва влияние на времето за зареждане и показване на информацията

### 2.2.3 Наличност

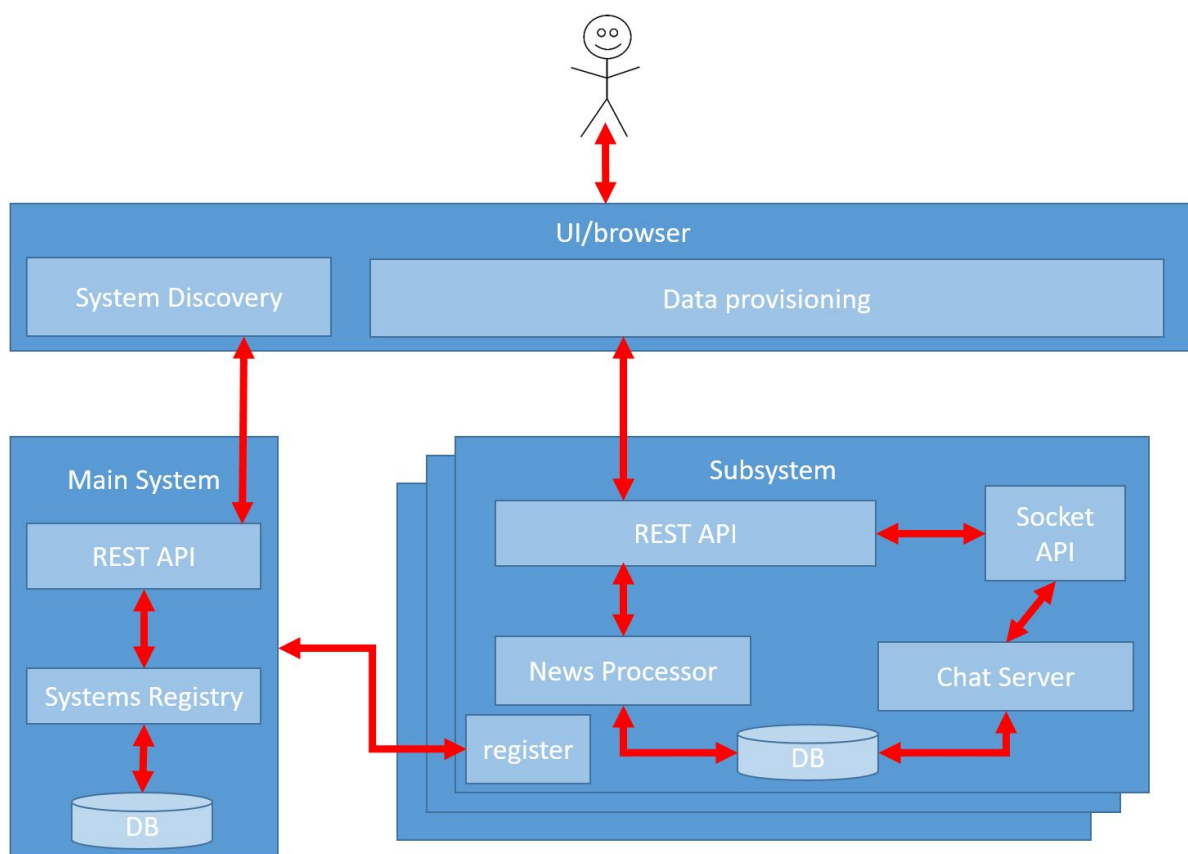
- Системата да е достъпна чрез Web Browser
- Да е налично android приложение, което да се поддържа от Android 7+
- Системата да е достъпна по всяко време на денонощието
- При проблем със системата времето за възстановяване на достъпа до нея да е не повече от 1 ден

### 2.2.4 Сигурност

- Съобразяване със законовите рамки в страните, където системата ще оперира (защита на потребителските данни)
- Ваксир на потребителската база в края на всеки ден
- Ваксир на новинарската база в края на всяка седмица

## 3 СИСТЕМНА АРХИТЕКТУРА

### 3.1 Архитектурно проектиране



Можем да разгледаме структурата на софтуерната система като няколко отделни компонента. Най-отгоре на графиката виждаме потребителя. Той ще използва системата през графичния интерфейс - web browser-а. Освен представянето на информацията и функционирането на чат системата, в тази първа част от системата ще става избора на университет или System Discovery. Идеята е че всеки университет представлява отделна подсистема със строго определен API който позволява хомогенна комуникация. Чрез избирането на университет се извика API на главната система и се осъществява връзката с подсистемата.

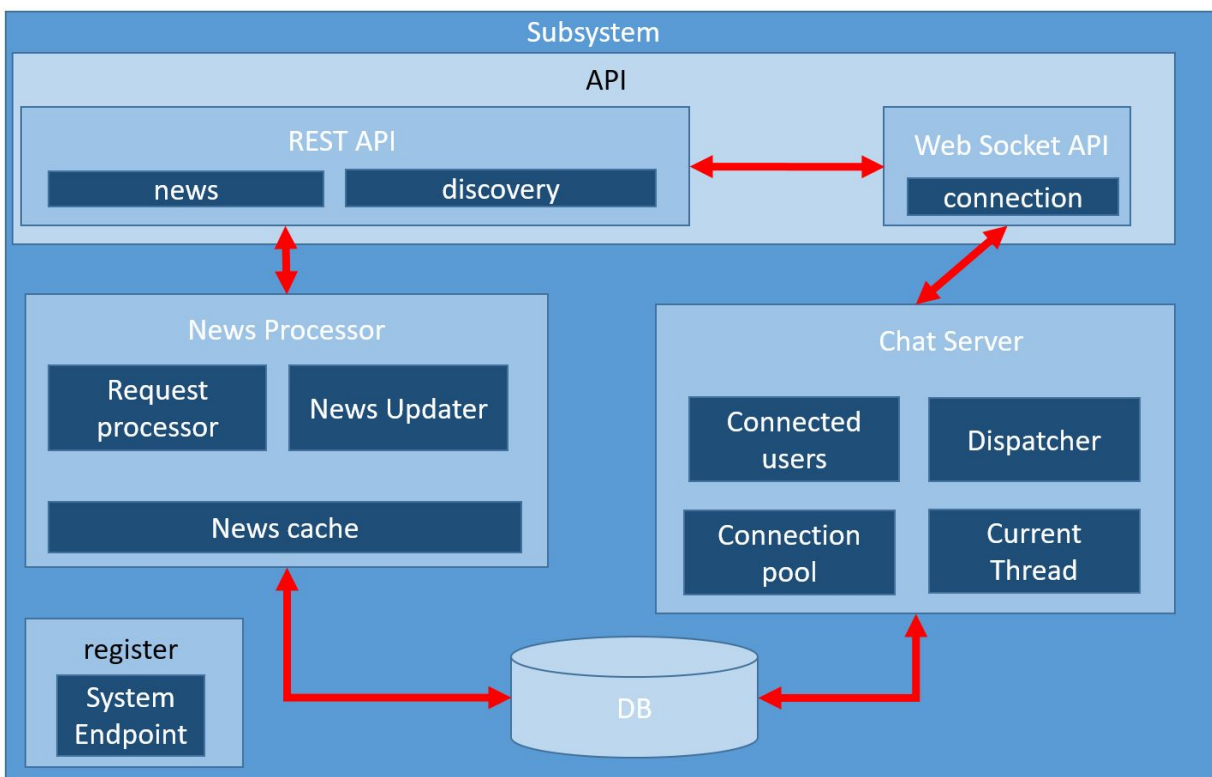
Втората част която ще разгледаме е Main System или главната система. Нейната цел е да бъде разпределител на връзките. А именно да знае кой университети/подсистеми са свързани в голямата софтуерна система и да предоставя API чрез който да се откриват



тези университети и връзките към тях. Сищо така тази система се грижи за регистрирането и проверката дали подсистемата е онлайн.

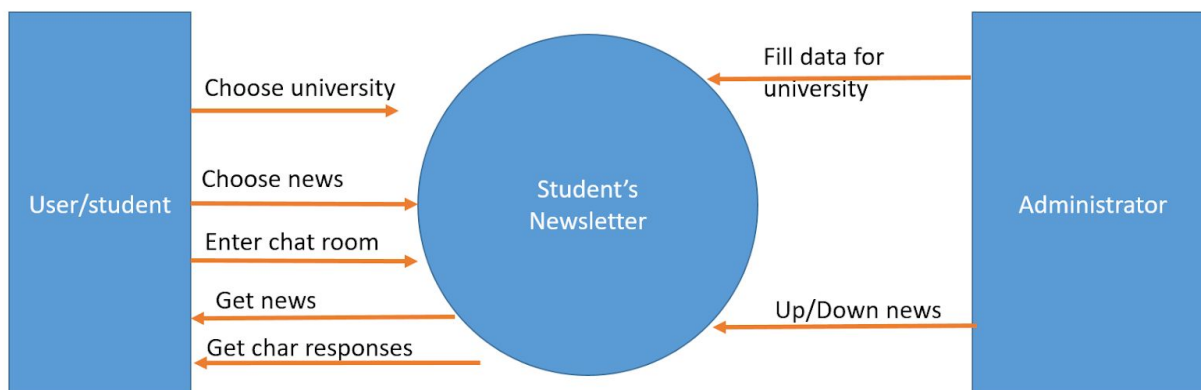
Третата част е самата подсистема или по-точно казано подсистеми. На графиката са показани като наслаждане на системи, като се има предвид, че са повече от 1. За подсистемите специфичното е че могат да бъдат имплементирани използвайки различни технологии. Единственото, което е задължително е да се използва унифициран API, който UI-а да може да консумира, без значение от подсистемата. Също така е важно socket API да бъде един и същ поради същите причини. Останалата част от системата може да бъде имплементирана по различни начини - може самата подсистема да е разпределена (да има един основен сървър load balancer и много подсървъри които да изпълняват заявките, но това е извън обхвата на проекта). На графиката подсистемата съдържа два компонента за REST API и за Web Socket API. Тяхната цел е да бъдат унифицирана връзка с UI-а. Те комуникират съответно със News Processor (процесорът на новини) - който се грижи за предоставянето на новините в определения формат и Chat Server (чат сървър) чиято цел е да поддържа чата в приложението като поддържа различните потребители, грижи се за техните сесии и да получават съобщенията от останалите участници в чата. Имаме и модул register (регистратор) - чиято цел е да регистрира подсистема в главната система и да предостави нужната информация.

## 3.2 Описание на декомпозицията



Както вече описавме по-горе архитектурата на подсистемата може да се променя. Единственото което трябва да бъде едно и също на различните системи е API, който те предоставят - а именно единен интерфейс, който превръща всяка подсистема като неразличима от останали без значение имплементацията. На горната графика е дадена архитектура на примерна подсистема - по-подробно разписана от горната. В нея можем да видим API layer-а (слоя) в който се съдържат двата основни типа - REST API Web Socket API. Като все от тях се грижи за различна част от функционалностите, които ще предоставя системата. В случая това са news, discovery и connection съответно, новините, откриването на системата и връзките за чат сървър.

Двата модула които се грижат за обработването и предоставянето на новите са News Processor и Chat Server. Процесорът на новини (News Processor) предоставя новините като може да бъдат имплементирани различни функционалности - например кеш, за да не се налага при четене всеки път да се извиква базата. Съответно news updater-а ще се грижи за обновяването на новините и консистентността на системата. Чат сървър (Chat Server) от своя страна ще предоставя възможността потребителите да обменят съобщения в общ чат, където всеки ще вижда какво публикува другия. Потребителите ще се subscribe-ват за конкретния час и след това ще бъдат известявани за нови съобщения.



Втората диаграма представлява проста data-flow диаграма. На нея са представени двете external entity (външни обекти), които са инициатори на данните - както публикуват така и консумират данни. В случая това са обикновения потребител(студент) и администратора на системата. Процеса който обработва данните е означен като Student's Newsletter - това е самата система.

Данните идващи/отиващи при обикновения потребител са изборите на университет и новини и съответно консумацията на новини и съобщения от чата. Данните на администратора са публикуване/редактиране на новини и данни за университетите.

### 3.3 Обосновка на проектирането

Архитектурните решения на системата са взети базирани на добрите практики за разработване на софтуер. Основната идея на екипа е системата да е гъвкава, да може лесно да се променя и да е скалируема. Гъвкавостта на системата се гарантира от

установените протоколи за комуникация- API. Те гарантират, че различните компоненти ще могат да комуникират и обменят данни, а тяхната имплементация е независима един от друг. Това позволява създаването на нови компоненти в различни срещи, използвайки различни технологии и езици за програмиране. По същия начин вече съществуващи системи ще могат да се интегрират като изградят този API слой, който да служи за комуникация. Скалируемостта може да се осъществява като се увеличават подсистемите и тяхната производителност или самите те бъдат проектирани така, че да бъдат разпределени. Избрали сме web browser да предоставя нашият интерфейс тъй като лесно може да се използва на различни видове устройства, това от своя страна не пречи да се създадат различни видове потребителски интерфейси консумиращи отвърдените API на системата - а именно мобилни приложения или десктоп приложения.

## **4 ДАННОВО ПРОЕКТИРАНЕ**

### **4.1 Описание на данните**

### **4.2 Речник на данните**

## **5 ПРОЕКТИРАНЕ НА ПОТРЕБИТЕЛСКИЯ ИНТЕРФЕЙС**

### **5.1 Обобщение на потребителския интерфейс**

Потребителите ще имат възможността да попълват форма за нова регистрация, както и такава за вход в системата с вече съществуващ акаунт.

След като са влезли в системата, потребителите имат достъп до страница с форма, позволяваща им да редактират своите имена и лични данни, както и да избират своя университет. Също така ще имат възможност да променят своята парола.

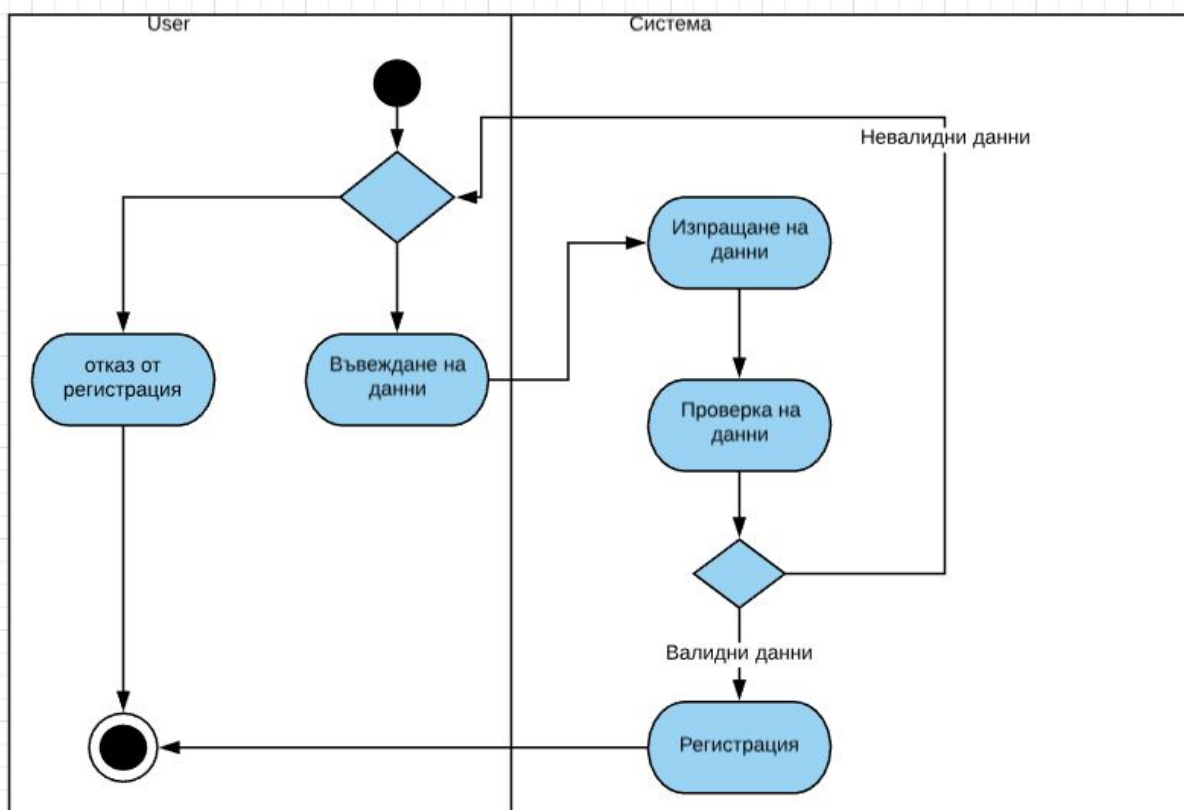
Потребителите, влезли в системата, имат достъп и до изглед, в който виждат последните новини. В този изглед има бутони за обновяване на списъка с новините и за преглед на по-стари такива.

Чрез кликане върху някоя от новините системата препраща потребителя към детайлен изглед на съответната новина, в който може да се прочете цялото ѝ съдържание.

Друг изглед за логнатите потребители е този с общия чат, в който динамично се визуализират съобщенията на другите студенти и има текстово поле за писане на съобщение и бутон за изпращането му.

За администраторите на системата съществува изглед с форма за добавяне на нова новина и определяне на това, дали тя да е видима за всички потребители или само за част от тях, както и за определяне на нейния приоритет.

## Регистрация

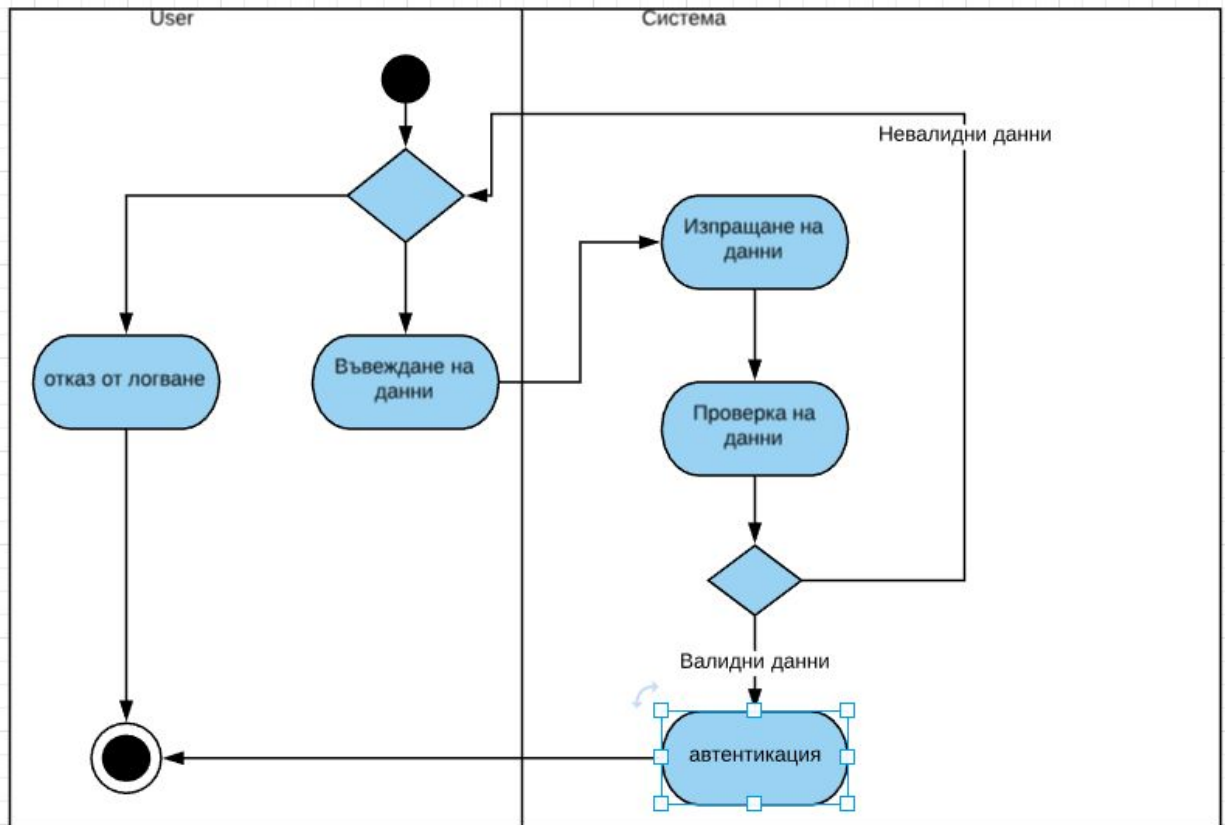


# Логване

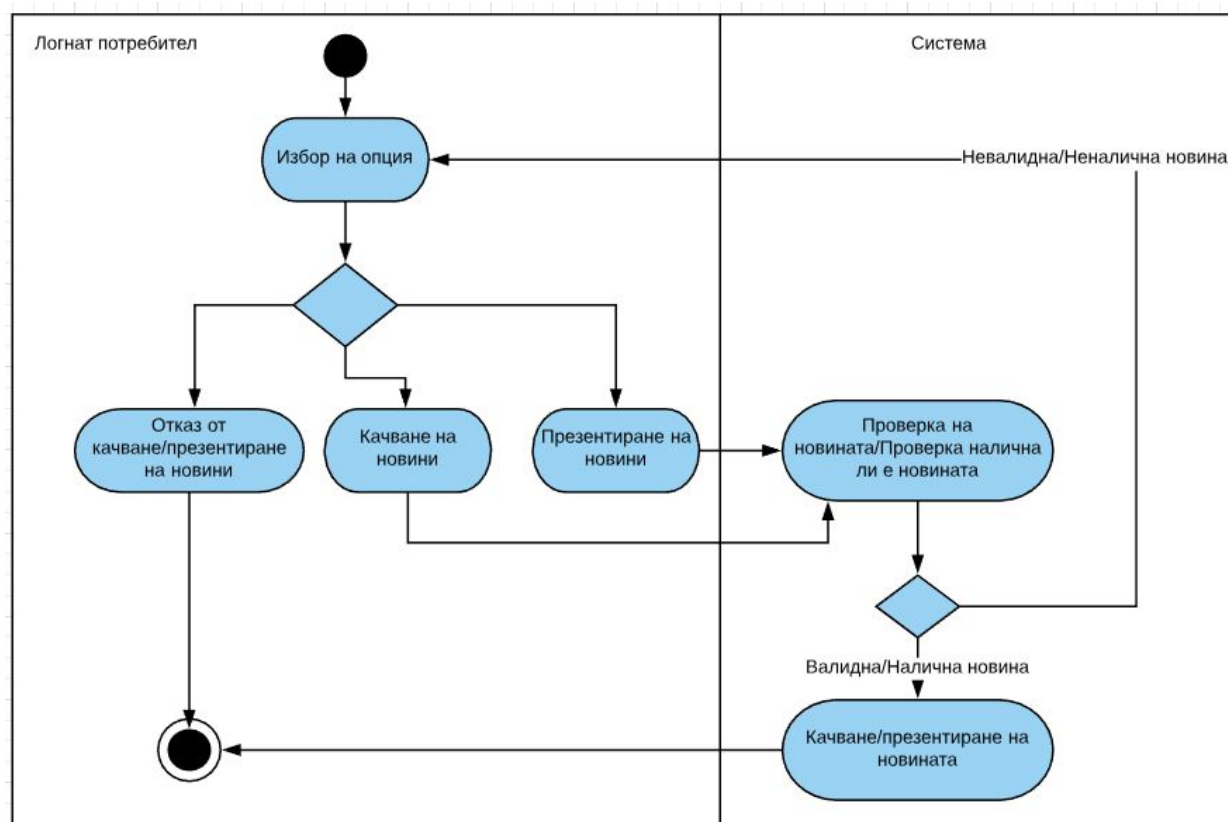
```
graph TD
    subgraph User
        Start(( )) --> Decision1{ }
        Decision1 --> Refuse([отказ от логване])
        Refuse --> End((( )))
        Decision1 --> Enter([Въвеждане на данни])
    end
    subgraph Система
        Enter --> Send([Изпращане на данни])
        Send --> Check([Проверка на данни])
        Check --> Decision2{ }
        Decision2 -- "Невалидни данни" --> Decision1
        Decision2 -- "Валидни данни" --> Auth([автентикация])
        Auth --> End
    end
```

The diagram illustrates the login process across two swimlanes: User and Система (System).

- User Swimlane:**
  - Starts at a solid black circle.
  - Flow to a decision diamond.
  - From the diamond, one path leads to the rounded rectangle "отказ от логване" (Login Refused), which then leads to the final bullseye state.
  - Another path from the diamond leads to the rounded rectangle "Въвеждане на данни" (Data Entry).
- Система (System) Swimlane:**
  - Receives data from "Въвеждане на данни".
  - Flow to "Изпращане на данни" (Data Sending).
  - Flow to "Проверка на данни" (Data Checking).
  - From "Проверка на данни", flow to a decision diamond.
  - If "Невалидни данни" (Invalid data), the flow loops back to the User's decision diamond.
  - If "Валидни данни" (Valid data), the flow leads to the rounded rectangle "автентикация" (Authentication).
  - From "автентикация", the flow leads to the final bullseye state.



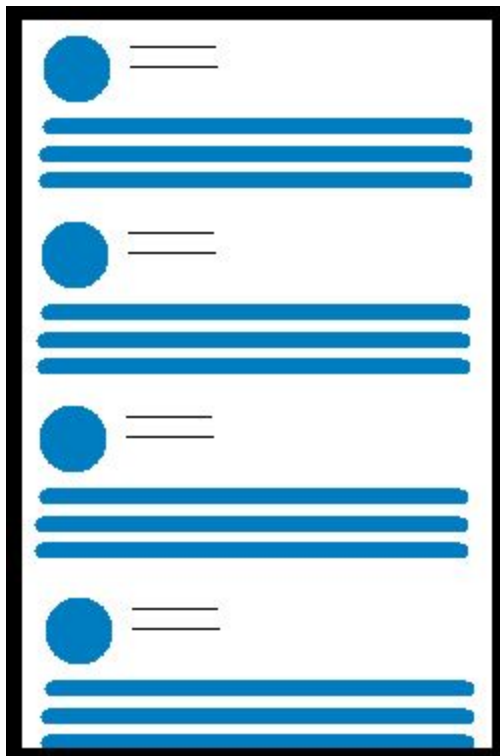
## Качване/Презентиране на новина



## 5.2 Екранни изображения



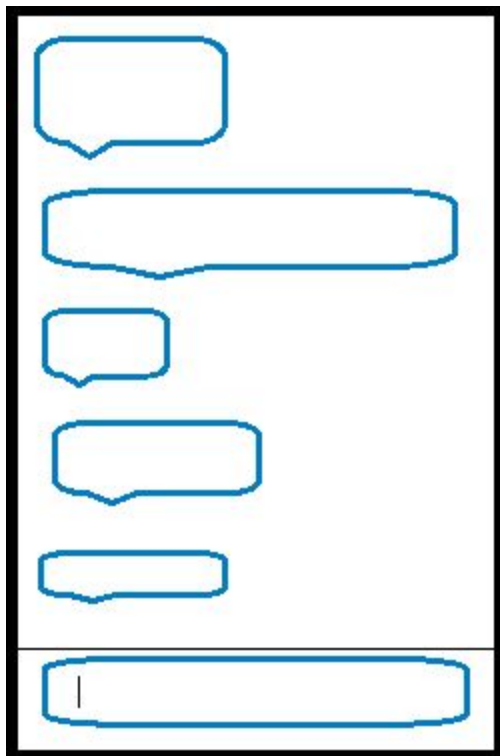
Фиг. 5.2.1 Изглед с новини в web среда



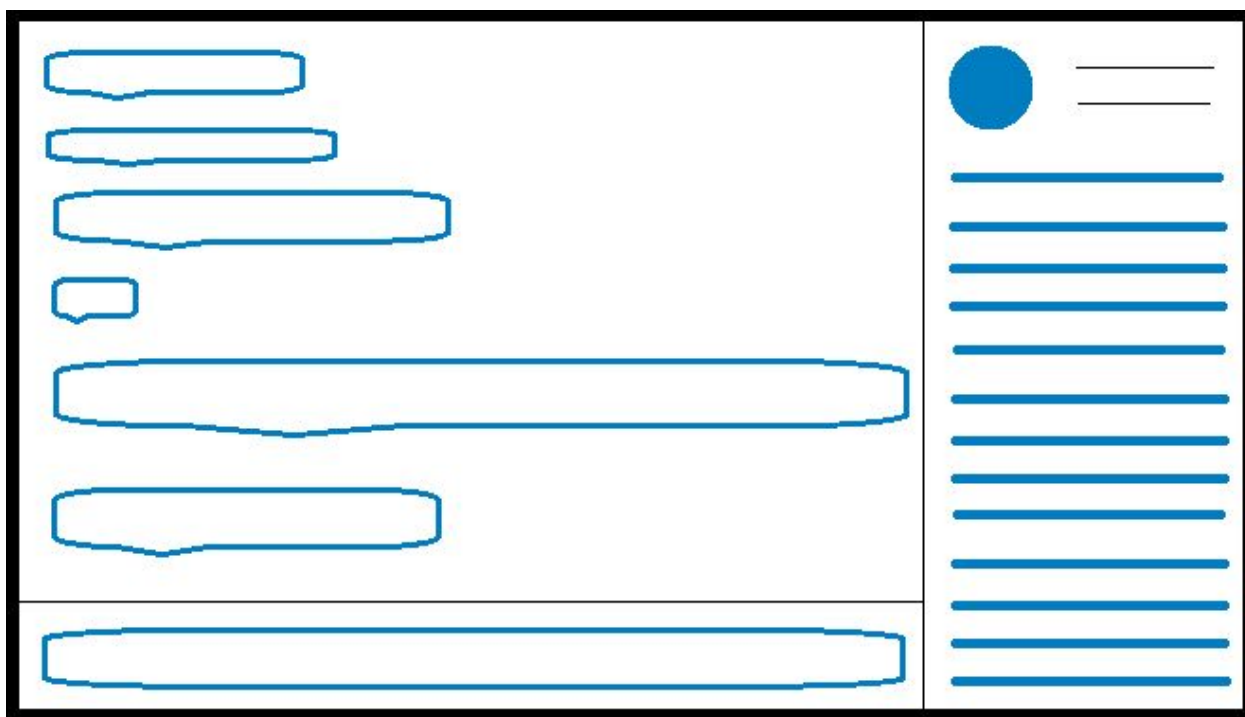
Фиг. 5.2.2 Изглед с новини в android среда

A vertical rectangular frame representing an Android screen. At the top, the text "Имена" (Names) is centered. Below it are two horizontal lines for input, followed by a solid blue horizontal bar. Further down, the text "Парола" (Password) is centered. Below this are three horizontal lines for input, followed by another solid blue horizontal bar.

Фиг. 5.2.3 Изглед с настройки за профила в android среда



Фиг. 5.2.4 Чат изглед в android среда



Фиг. 5.2.5 Чат изглед в web среда



## 6

## МАТРИЦА НА ИЗИСКВАНИЯТА

[illegible]