The Implementation of an Online Shopping Assistant Bot With Messenger

By: Iyanu O. Adelekan
Matriculation Number: 179088
Supervisor: Dr. O.F.W Onifade

Introduction

- Since the advent of internet powerhouses like Amazon, the online shopping market has witnessed unforeseen proliferation in user interest and consumption.
- In 1999, Internet sales were estimated to be more than double that of 1998, $20 billion (Forrester Research).
- Despite the indisputable growth e-commerce has witnessed, research indicates that consumer e-commerce sales currently account for less than 1% of total retail sales (Wolfinbarger, 2000).
- There is currently no estimated upper limit to the growth of the e-commerce sector.

Problem Statement

As the e-commerce market proliferates, more efficient means for consumer item discovery are developed. Regardless these developments, e-commerce accounts for less than 1% of total retail sales. Millions of consumers are still unable to discover desired items effectively because we do not have a very simple, user-friendly and <u>highly efficient</u> means to find items online.

Aim and Objectives

The aim of this project is to solve the problem of item discovery in e-commerce by designing and implementing an intelligent online shopping assistant chatbot to aid consumers to efficiently discover and purchase products.

Three objectives stem from this aim. These objectives are outlined below:

- To create an enjoyable, consumer centric, shopping experience based on chat.
- To design and implement an intelligent Facebook messenger bot with integrated artificial intelligence.

Literature Review

S/N	AUTHOR(S) AND WORK	YEAR AND SOURCE	GOAL REACHED	Remark
1	David Gefen, Building Effective Online Marketplaces with Institution Based Trust	2004, Information systems research 15 (1), 37-59	Integrates sociological and economic theories to propose that the perceived effectiveness of institutional mechanisms engender buyer trust in the community of online sellers.	This work gives a valid proposition about the effectiveness of institutional mechanisms in fostering buyer trust.
2	Sirini S. Srinivasan, Rolph Anderson, Kishore Ponnavolu, Customer loyalty in e-commerce: an exploration of its antecedents and consequences	2002, Journal of retailing, volume 78, issue 1	Identifying eight factors that potentially impact e-loyalty and develop a scale to measure these factors.	The factors identified by this work will benefit online shopping solutions in promoting customer loyalty.
3	Douglas L. Peckover, Intelligent agents for electronic commerce	September 12, 2000, BiBTeX, US 6119101 A	Developing a system for electronic commerce having personal agents that represent consumers and providers in a virtual marketplace.	This work gives a definitive assessment of the applications of intelligent agents to the e-commerce sector.
4	Robert Guttman, Alexandros Moukas, Pattie Maes, Agents as mediators in e-commerce	1999, Springer, Berlin, Heidelberg	Established the fact that intelligent agents are useful for the information-rich and process-rich environment of electronic commerce.	This work shows how useful intelligent agents can be in developing information rich e-commerce solutions.

Methodology: Tools utilised

- Programming languages: Javascript and Ruby
- Integrated development environments: WebStorm and RubyMine
- Web frontend styling and markup: HTML and CSS
- Frameworks: Bootstrap, ExpressJS, Ruby on Rails
- Chatbot platform: Facebook Messenger
- Artificial intelligence: API.AI

System Requirements

Functional requirements:

- System must be able to authenticate users.
- Wishlist creation must be facilitated.
- System must be able to accept payments.
- User budget tracking and transaction history retrieval must be supported.
- Product discovery tasks must be supported.

Non-functional requirements:

- The system must have zero downtime.
- The upper bound latency of server responses must be 40ms.
- The system must adhere to industry set security standards.
- Chatbot UI must be clean and efficient.

Architecture Diagram of The System
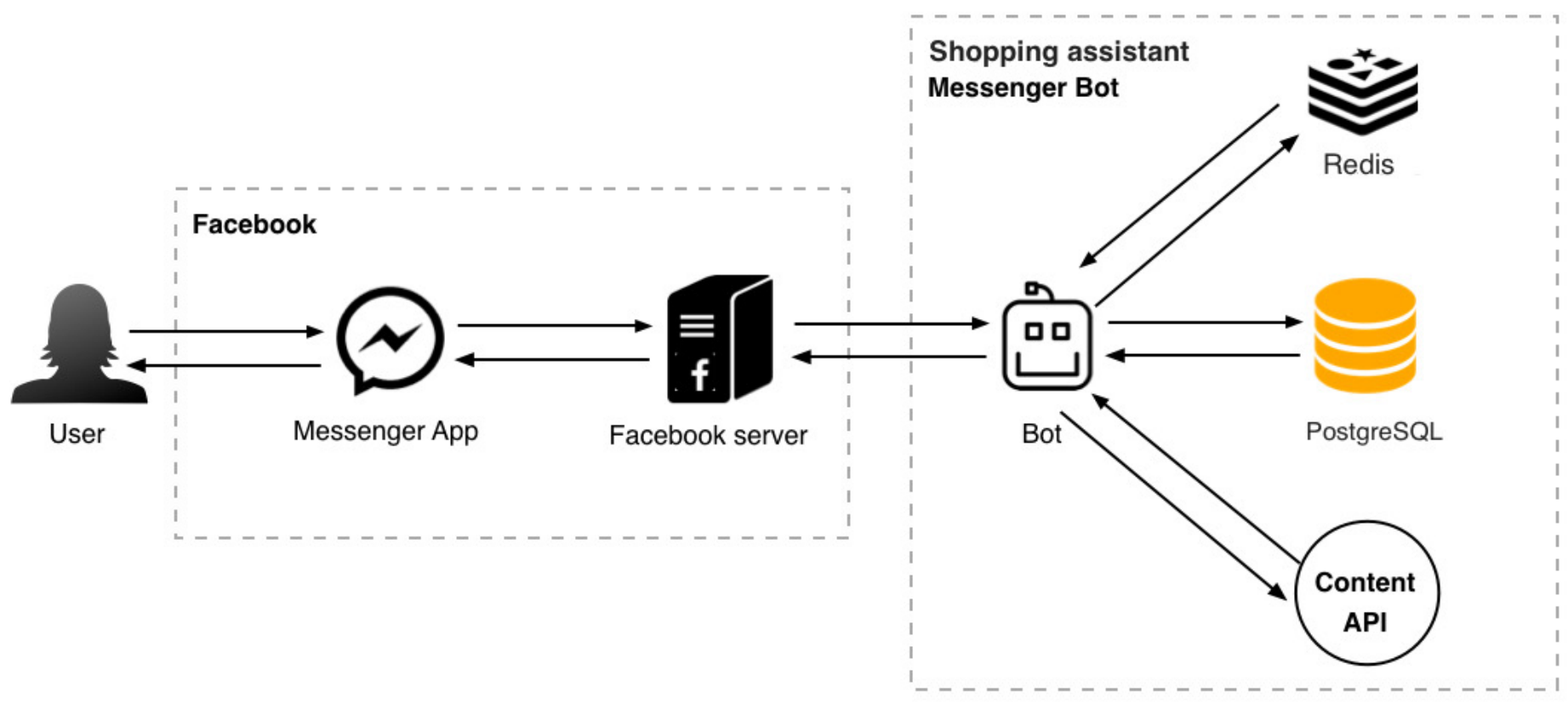
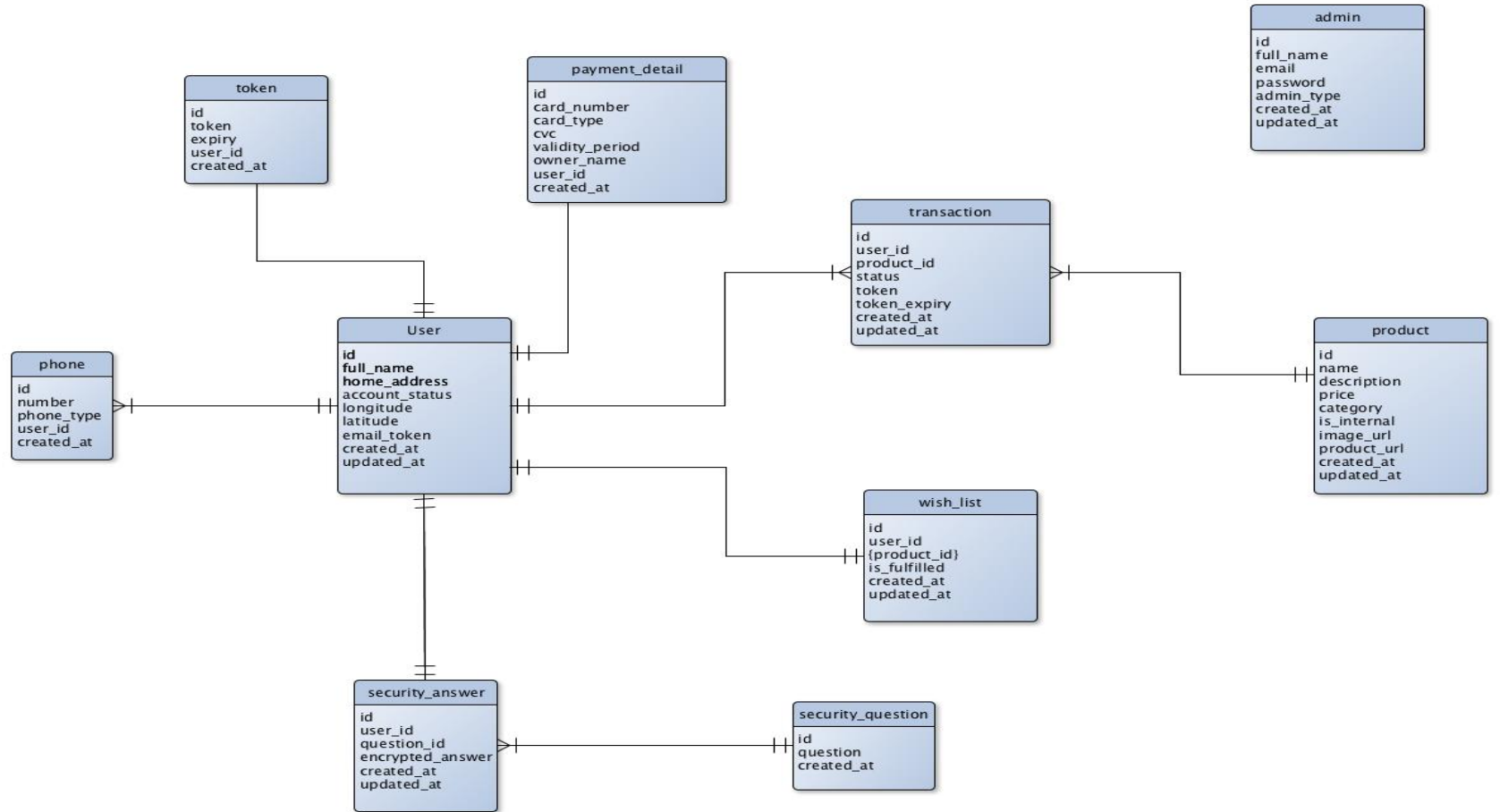# Typical Interactions Between Shopping Assistant Subsystems
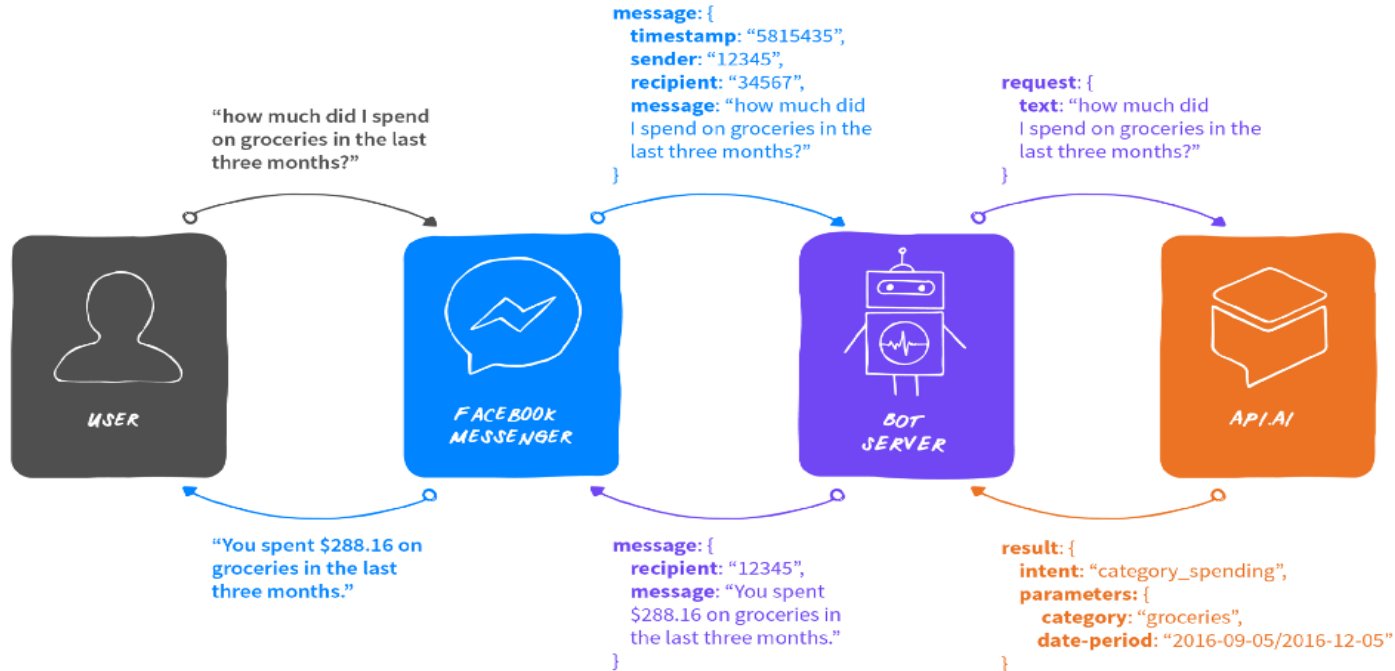
Context Diagram
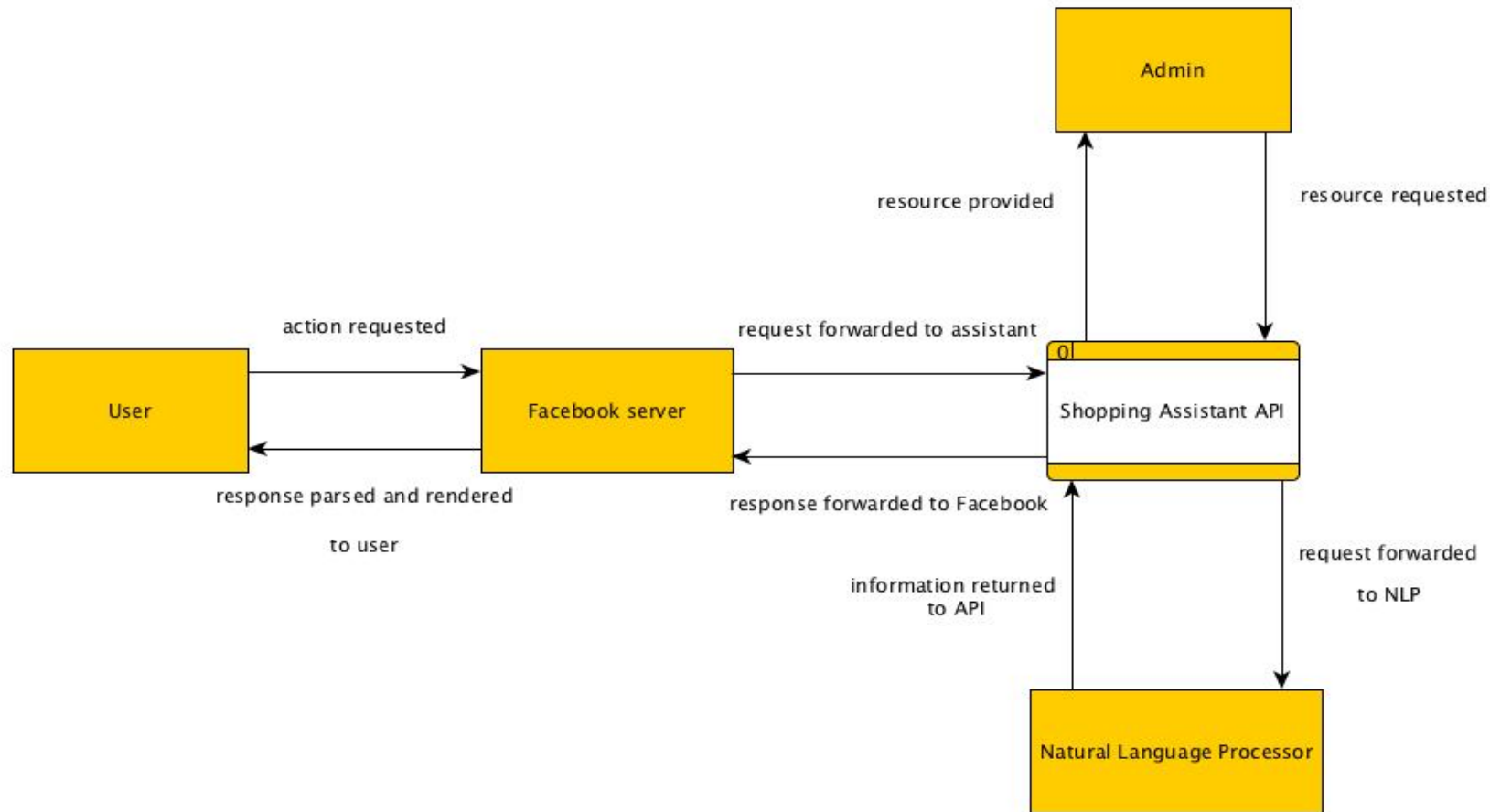
Data Flow Diagram: Level 0
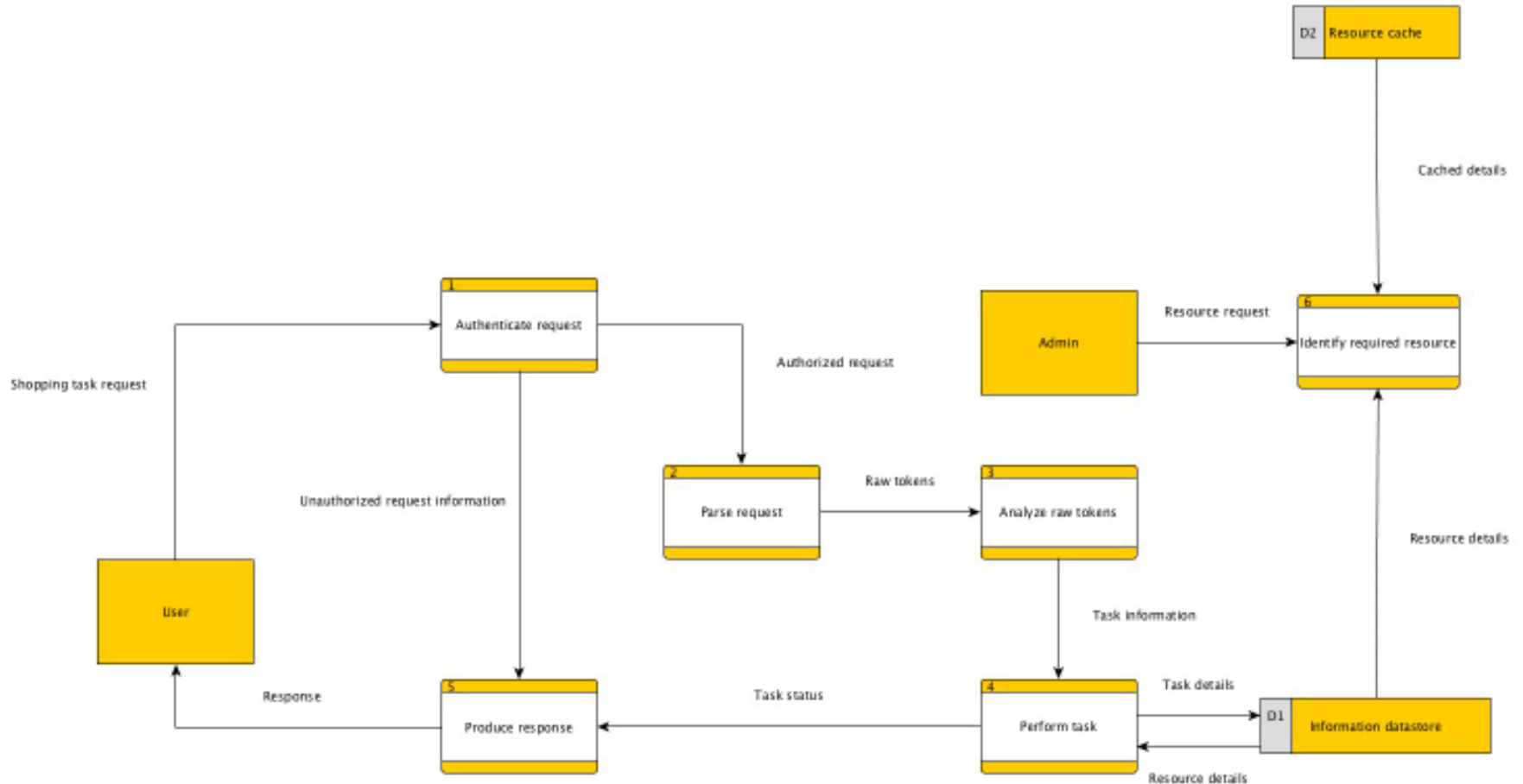
E-R Diagram

Use Case Diagram

Online shopping assistant system

- Request product search
- View transaction history
- Retrieve wish list
- Modify wish list
- Find store
- Order product
- Confirm transaction
- Cancel transaction
- Update account credentials
- Suspend user
- View users
- View transactions

User

Admin

Relevance of Study

The concluded work alleviates difficulties experienced by online shoppers with respect to online item discovery.

The implemented system helps to automate tedious manual processes involved in online shopping.

In addition, this concluded work empowers e-commerce consumers to spend more time on activities critical to the accomplishment of their shopping related goals.

Source Code

Finding a store

```javascript
/**

* Used to search for a store based on provided details

* @param name - name of the store to find

* @param city - city in which the store is located

* @param country - country in which the store is located

* @param productName - name of product to find

* @returns {Promise<any>}

*/

prototype.findStore = (name, city = null, country = null, productName = null) => {

  return new Promise(((resolve, reject) => {

    if (name === '') {

      Store.findAll()

        .then(stores => resolve(stores))

        .catch(reason => {

          console.log(reason);

          reject(reason)

        })

    } else {

      if (city != null && country != null && productName != null) {

        Product.findAll({

          where: {

            name: {

              [Op.iLike]: `%${productName}%`

            }}}).then(products => {

          let data = [];

          let significantProducts = products.filter(product => {

            let store = product.store;
```

```javascript
                    return store.name === name && store.city === city && store.country
=== country

                });

                significantProducts.forEach(product => data.push(product));

            resolve(data)

        }).catch(reason => {

            console.log(reason);

            reject(reason)

        });

    } else if (city != null && country != null) {

        Store.findAll({ where: { name: name, city: city, country: country }})

            .then(stores => resolve(stores))

            .catch(reason => {

                console.log(reason);

                reject(reason)
        })

    } else if (city != null) {

        Store.findAll({ where: { name: name, city: city }})

            .then(stores => resolve(stores))

            .catch(reason => {

                console.log(reason);

                reject(reason)

        })

    } else if (country != null) {

        Store.findAll({ where: { name: name, country: country }})

            .then(stores => resolve(stores))

            .catch(reason => {

                console.log(reason);

                reject(reason)
```

Creating a new product

```javascript
            })

        } else {

            Store.findAll({ where: { name: name }})

                .then(stores => resolve(stores))

                .catch(reason => {

                    console.log(reason);

                    reject(reason)

                })

            }

        }

    }))

};
```

```javascript
prototype.createProduct = (
    name,
    description,
    price,
    quantity,
    isInternal,
    brand,
    colour,
    storeId,
    categoryId,
    imageUrl = null,
    productUrl = null) => {

    return new Promise(((resolve, reject) => {
        Product.create({
            name: name,
            description: description,
            price: price,
            quantity: quantity,
            internal: isInternal,
            brand: brand,
            colour: colour,
            storeId: storeId,
            categoryId: categoryId,
            imageUrl: imageUrl,
            productUrl: productUrl
        }).then(product => {
            resolve(product)

        }).catch(reason => {
            console.log(reason);
            reject(reason)
        })
    }))
};
```

Result

- The output of the implementation stage is a fully functional automated shopping assistant chatbot.
- Full documentation of the system in form of a user manual and source code documentation is another tangible output which came out of the system design and implementation stages.
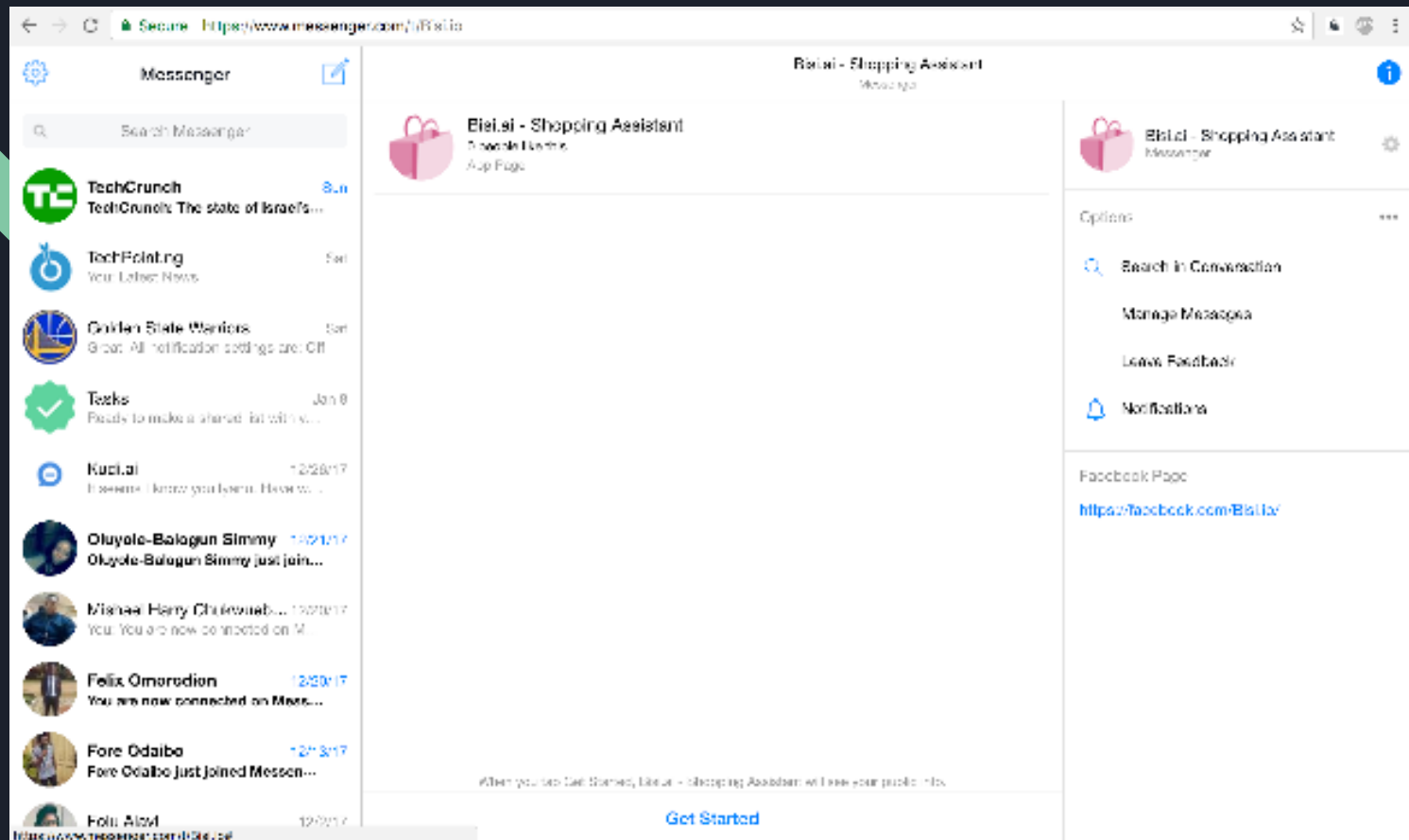
Figure 7: Starting a conversation

User onboarding
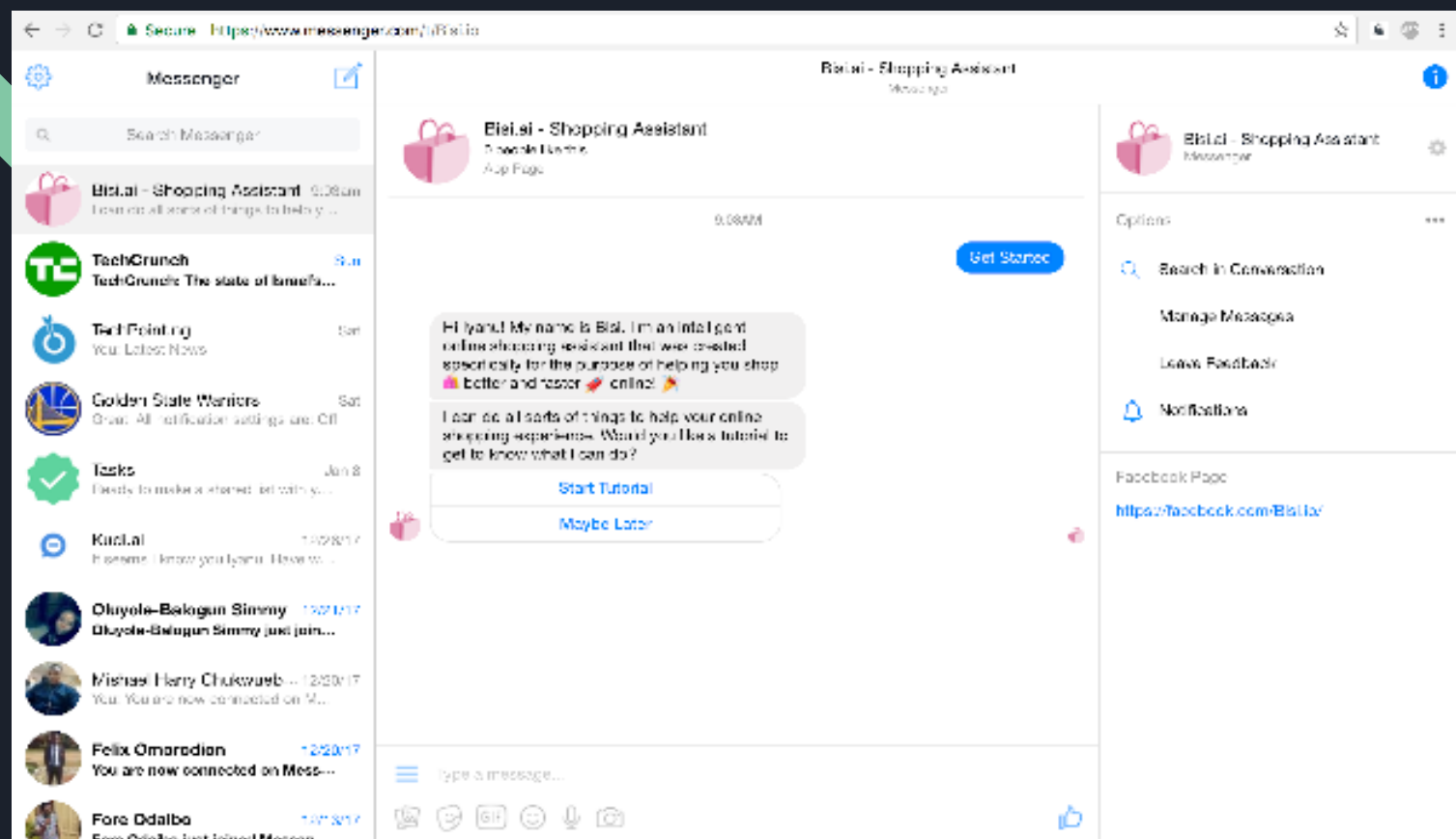
Figure 8: Bot requesting to put a user through the tutorial process
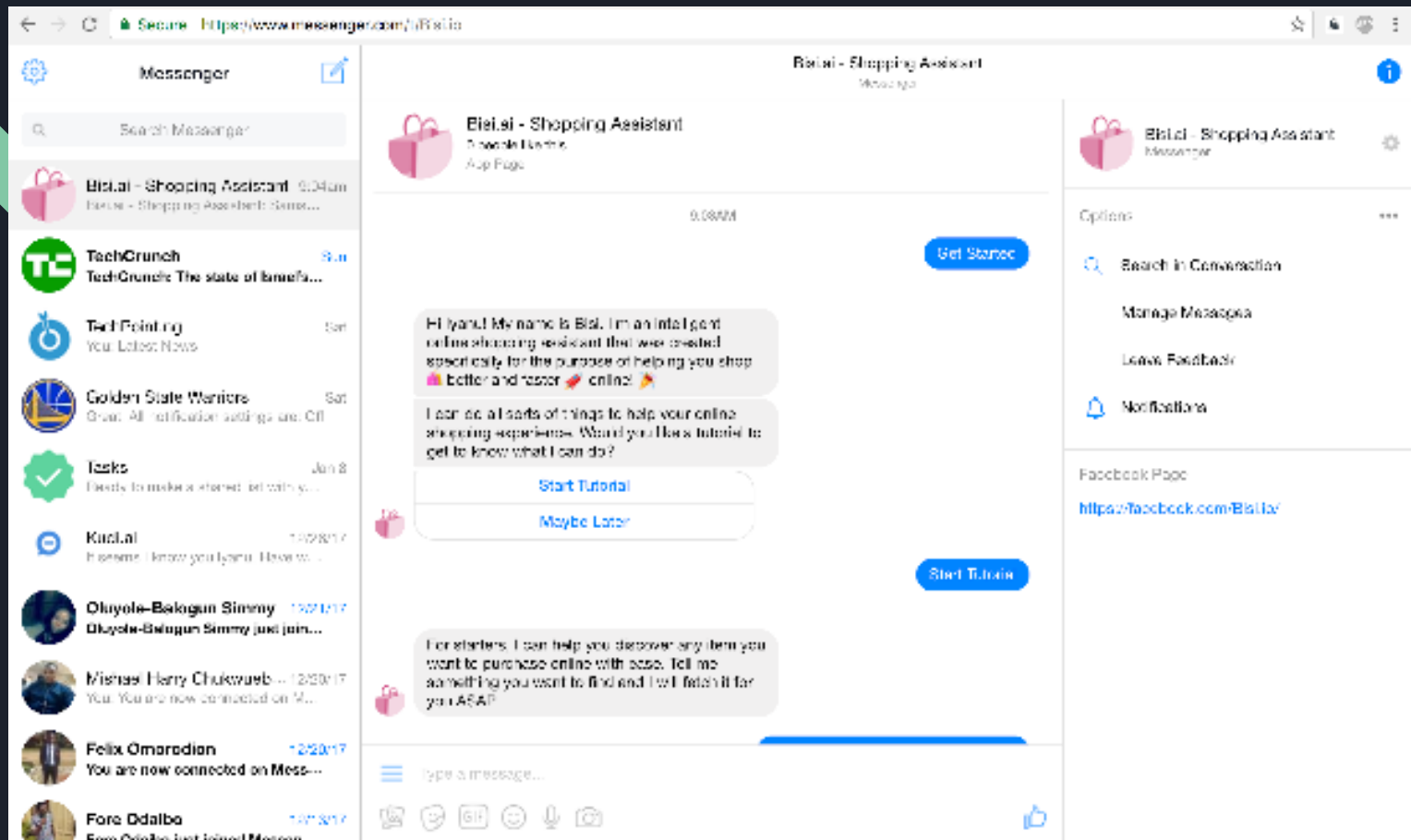
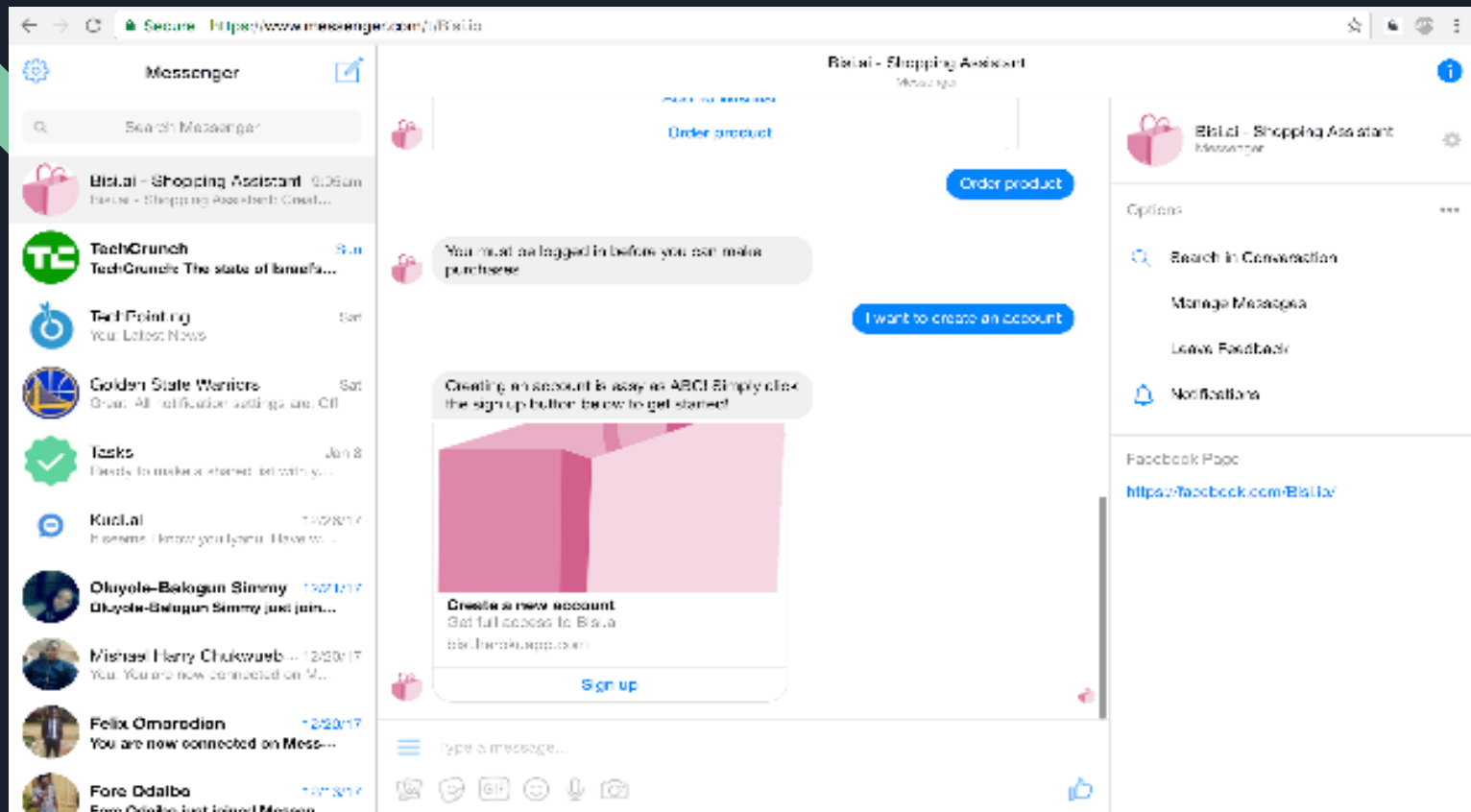**Figure 9:** Bot putting user through tutorial

Account creation

Figure 10: Shopping assistant obliging an account registration request
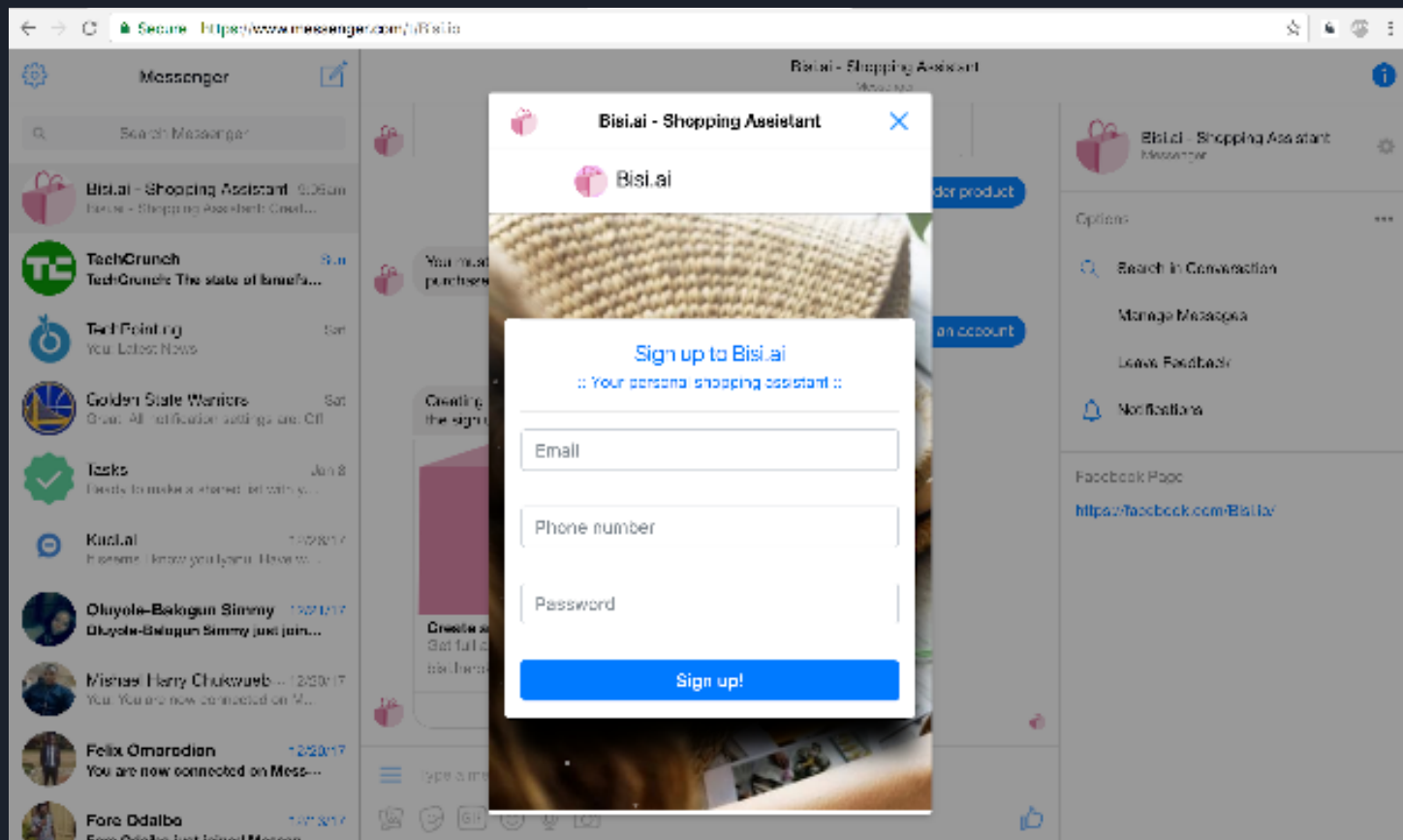
Figure 11: Account registration web view

Account login
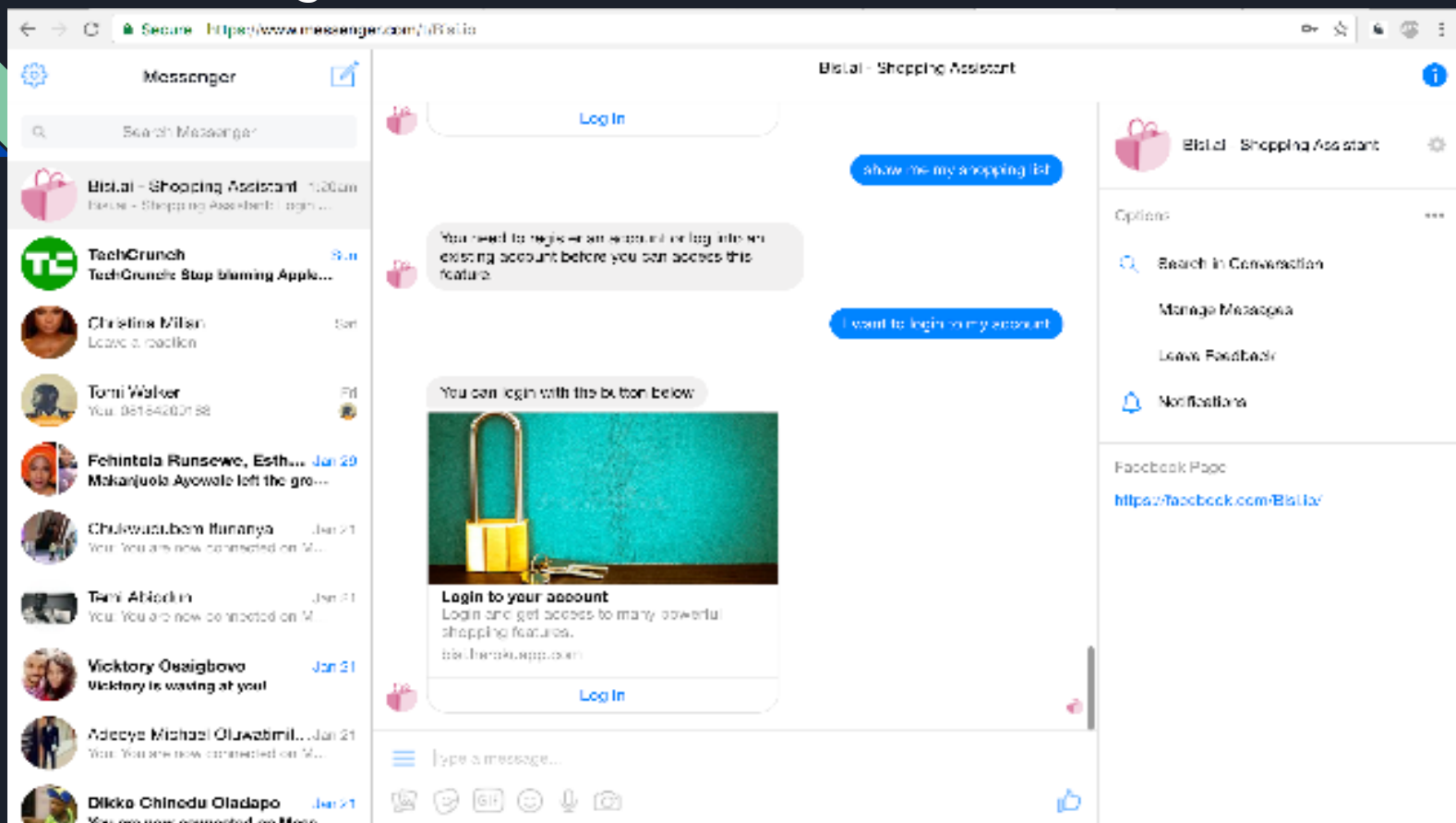
Figure 12:   Shopping assistant acknowledging a login request
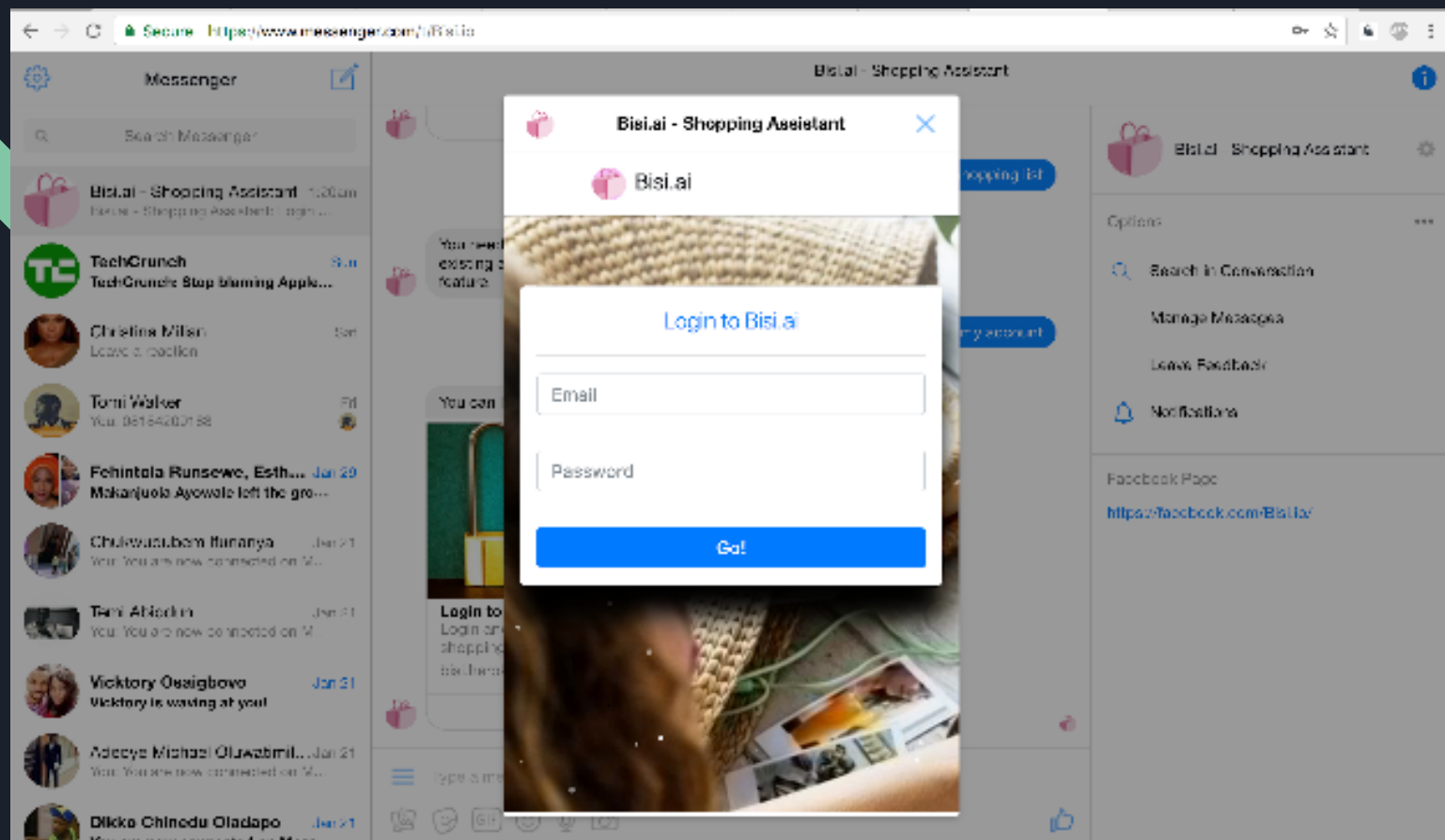
Figure 13: Account login web view

Shopping List Maintenance
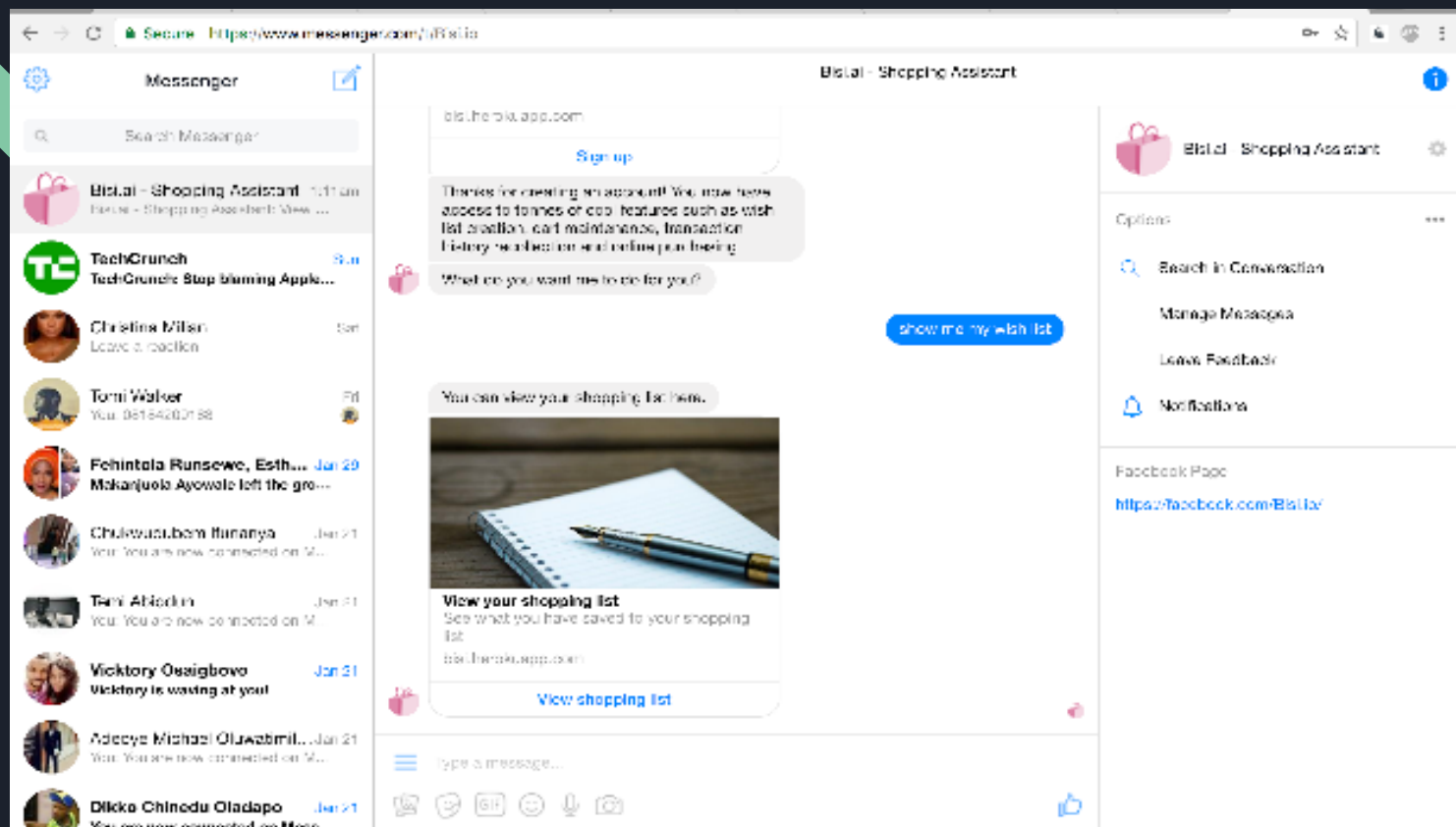
Figure 14: Shopping assistant facilitating access to a user's shopping list
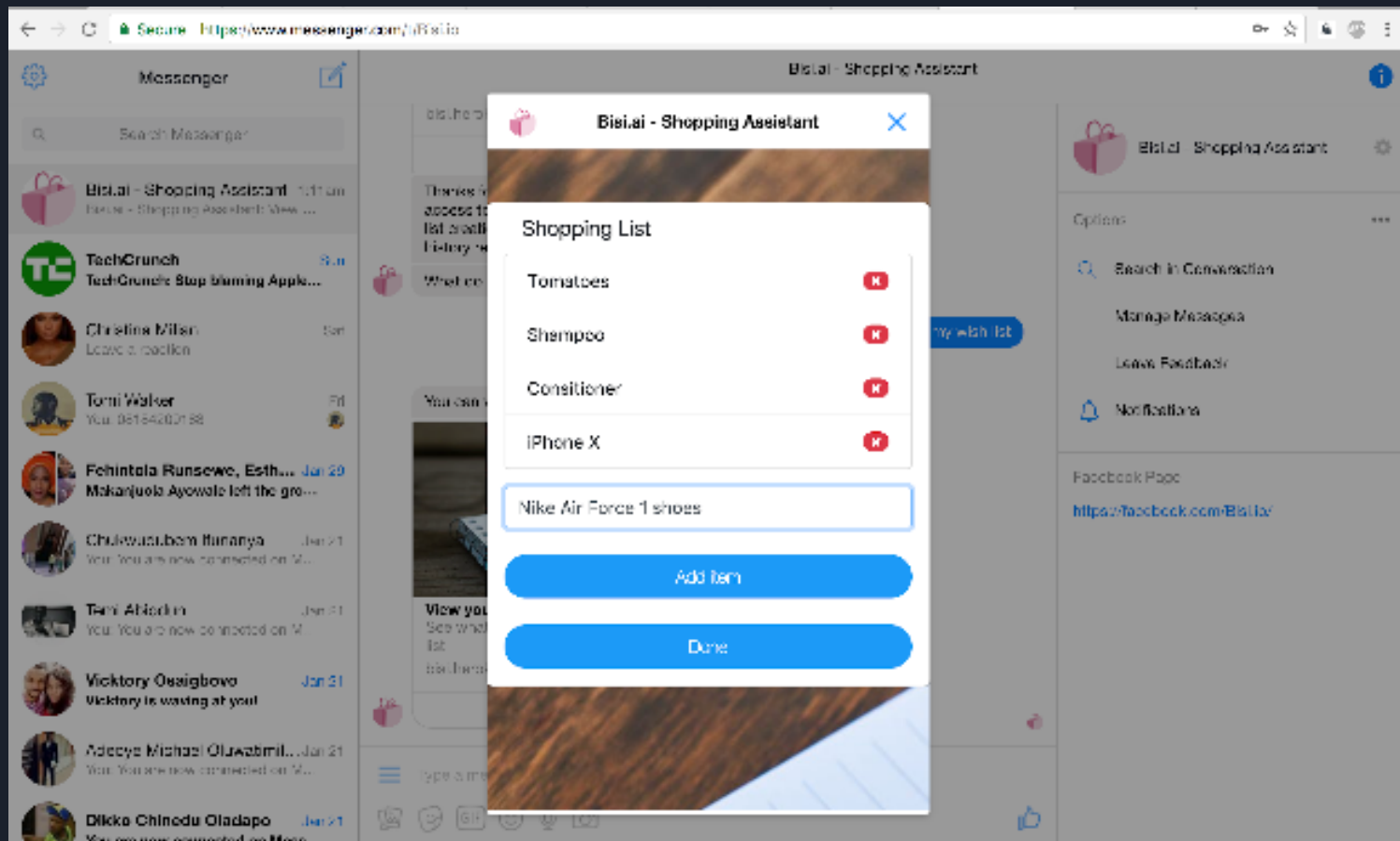
Figure 15: User adding items to his shopping list

Online and Offline Item Discovery
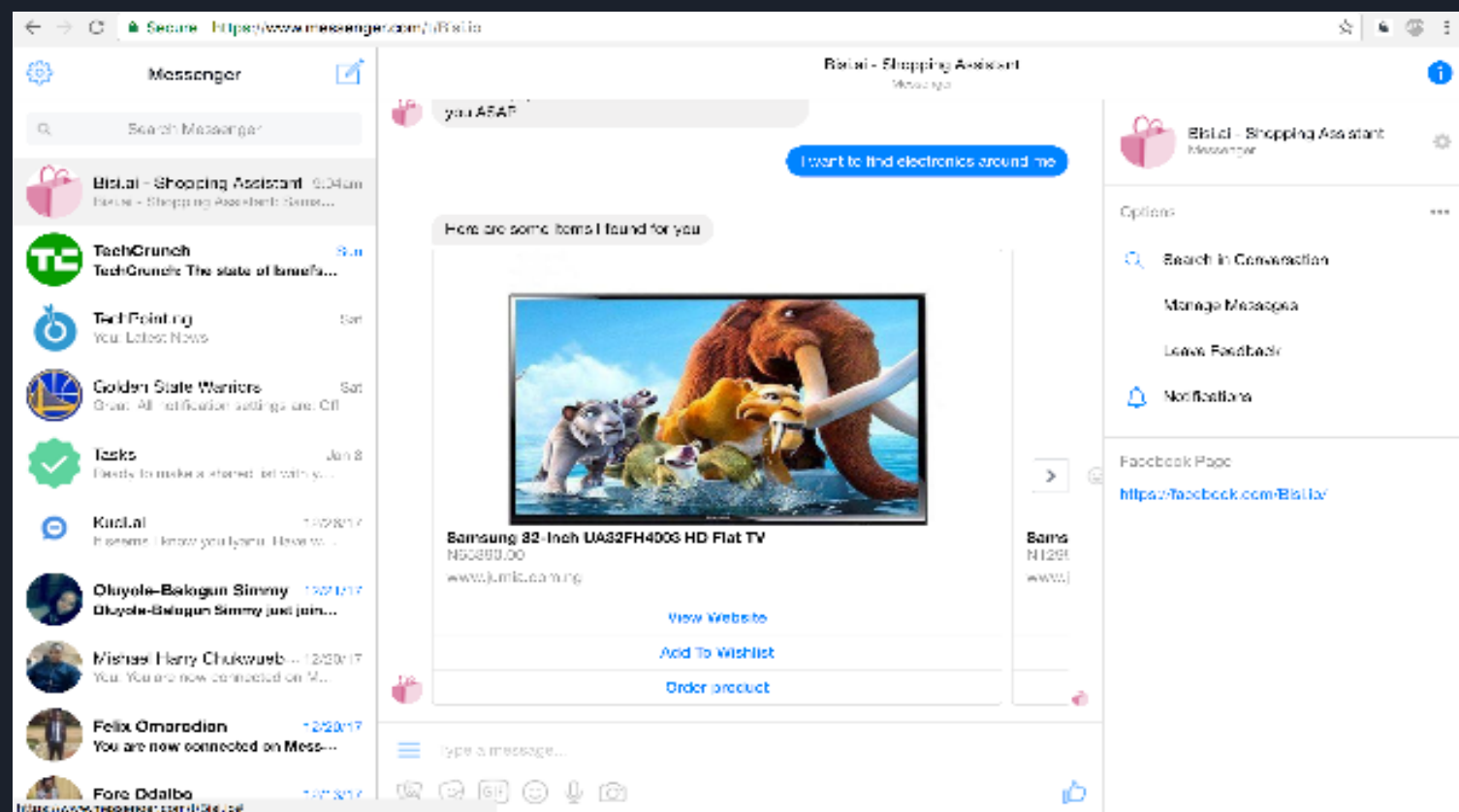
Figure 16:   Bot seamlessly discovering an item for a user
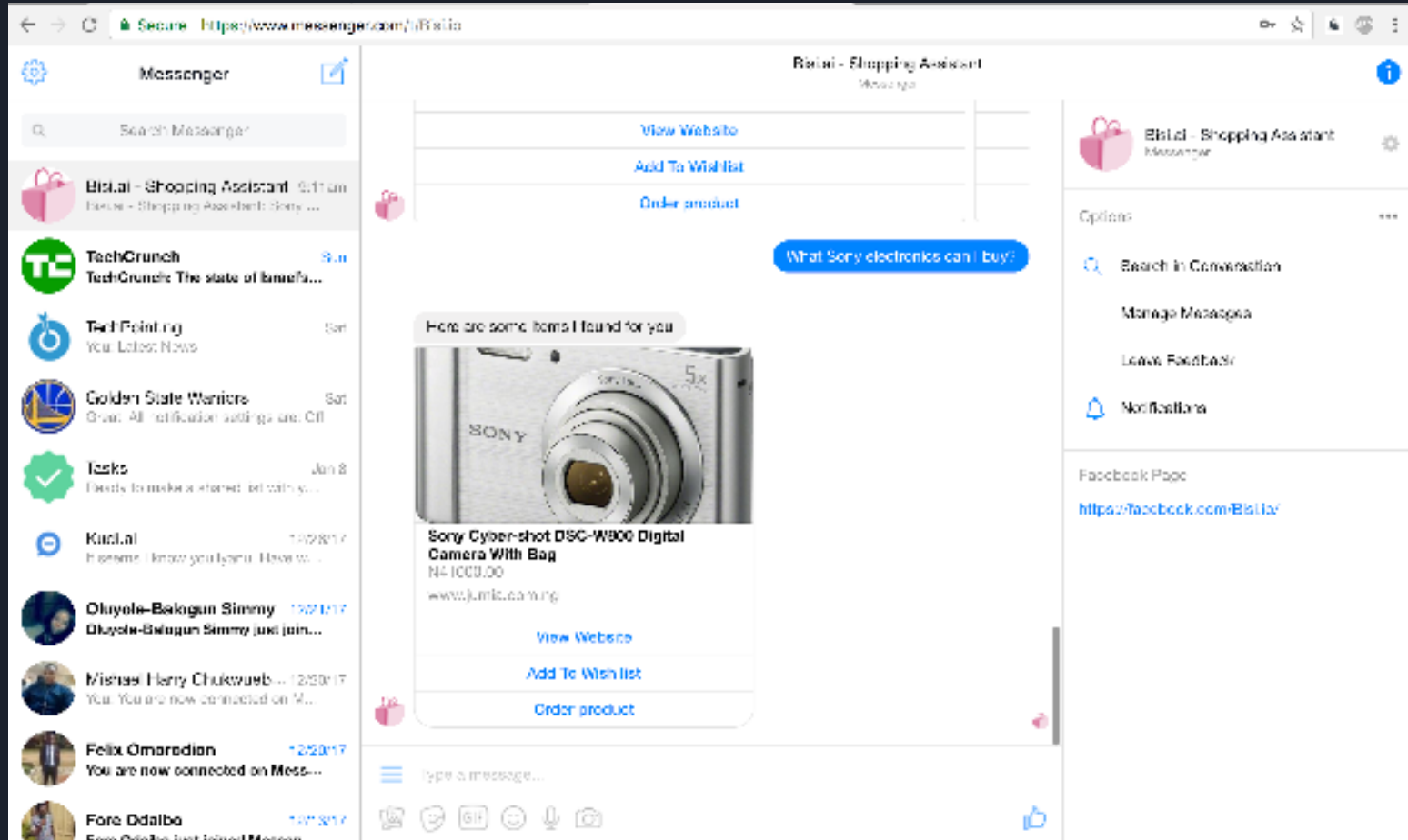
Figure 17:   Bot servicing a search query for a specific brand of product

Item Purchase
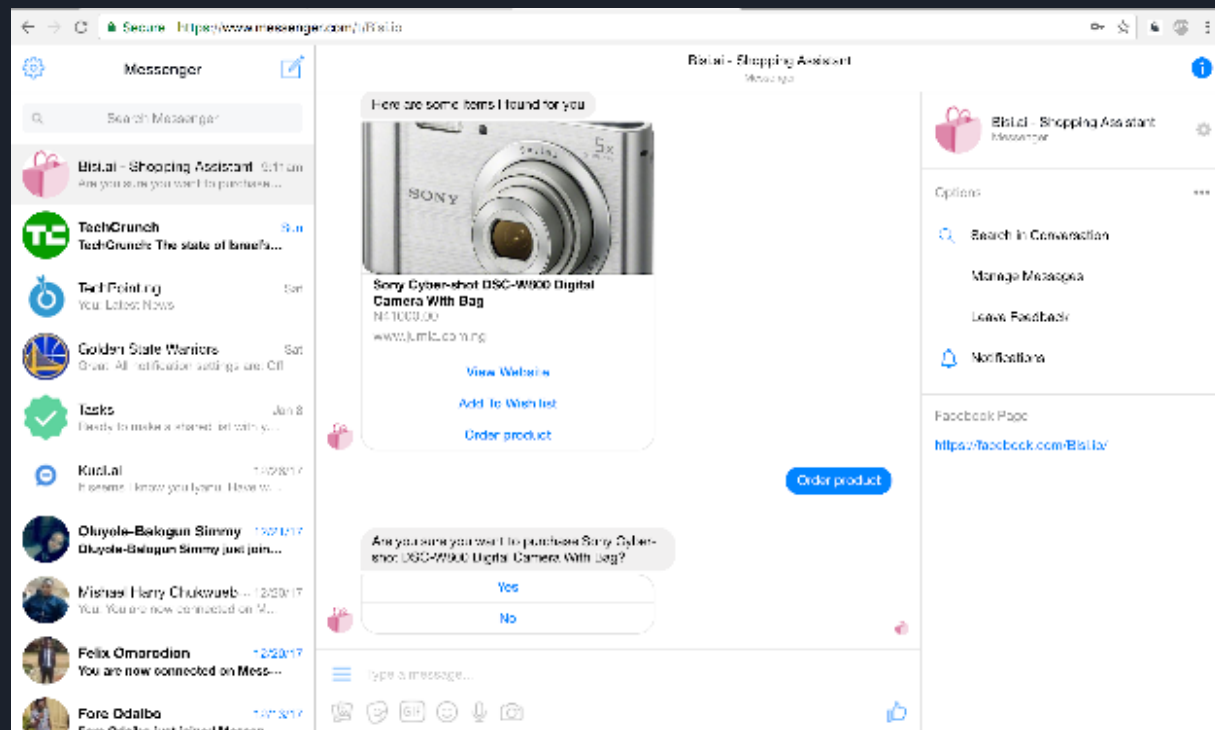
Figure 18:   Initiation of product purchase request
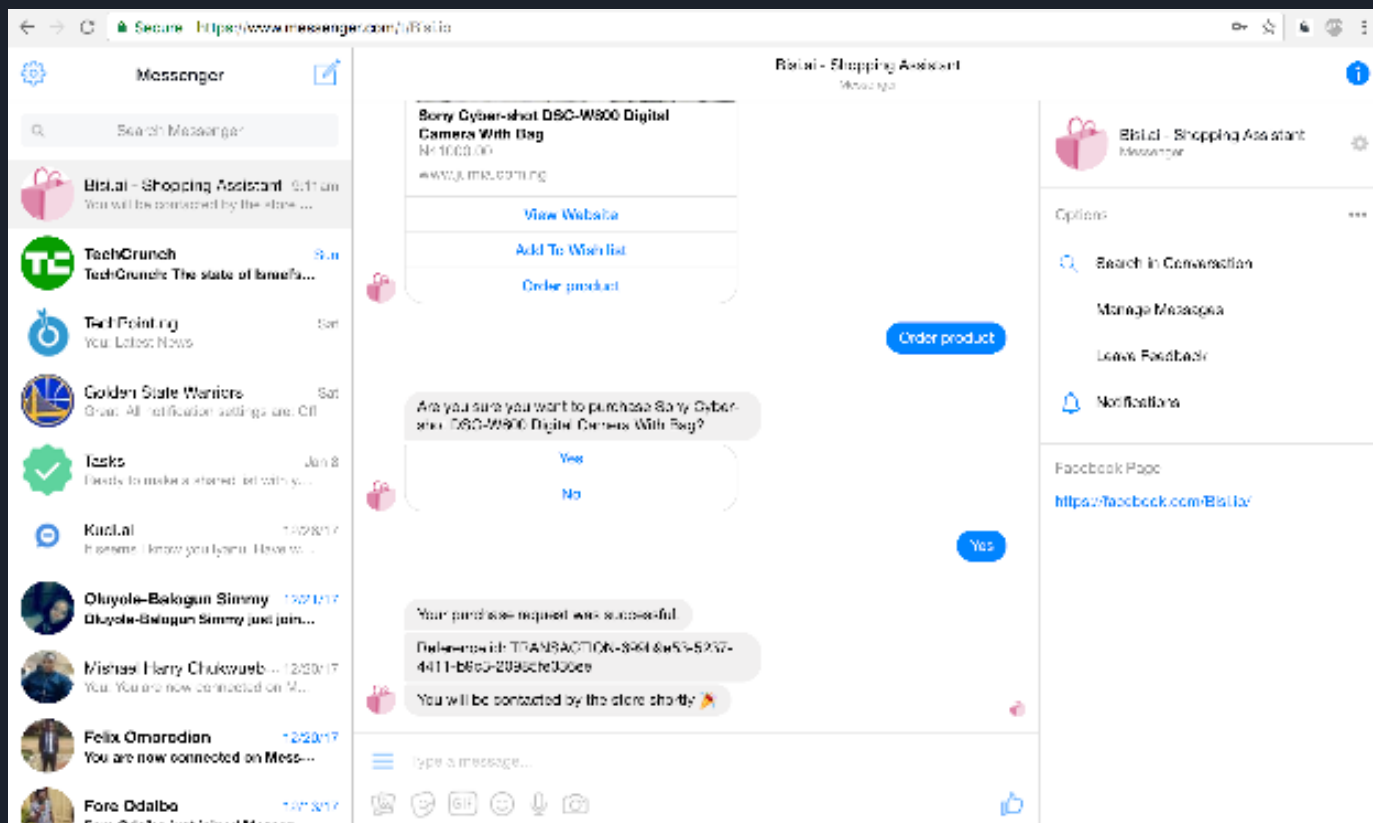
Figure 19: Completion of purchase request
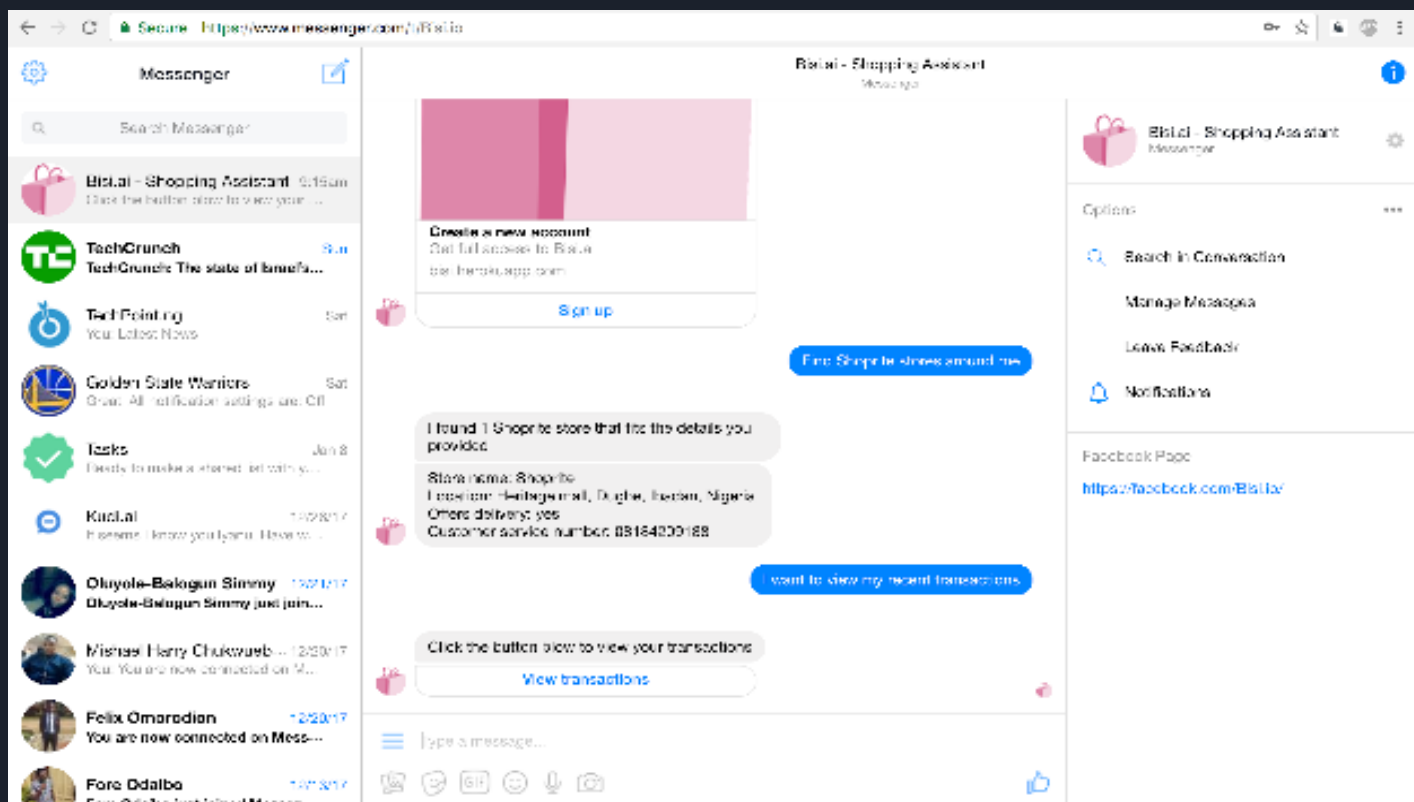
Transaction History Maintenance

Figure 20:   Shopping assistant presenting user with a widget for the viewing of transaction history

Figure 21: User viewing his transaction history
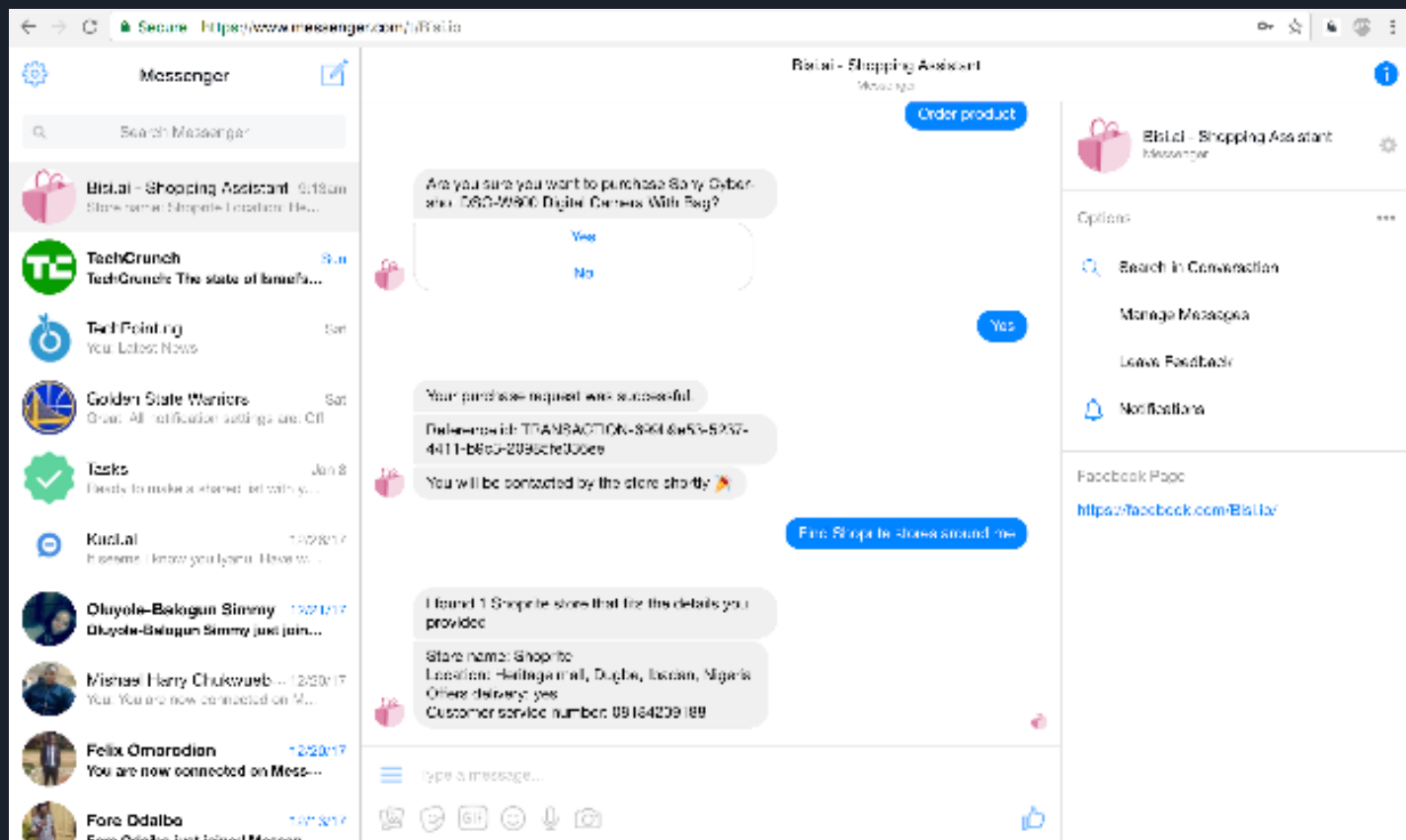
Store Discovery

Figure 22: Shopping assistant discovering stores around a user
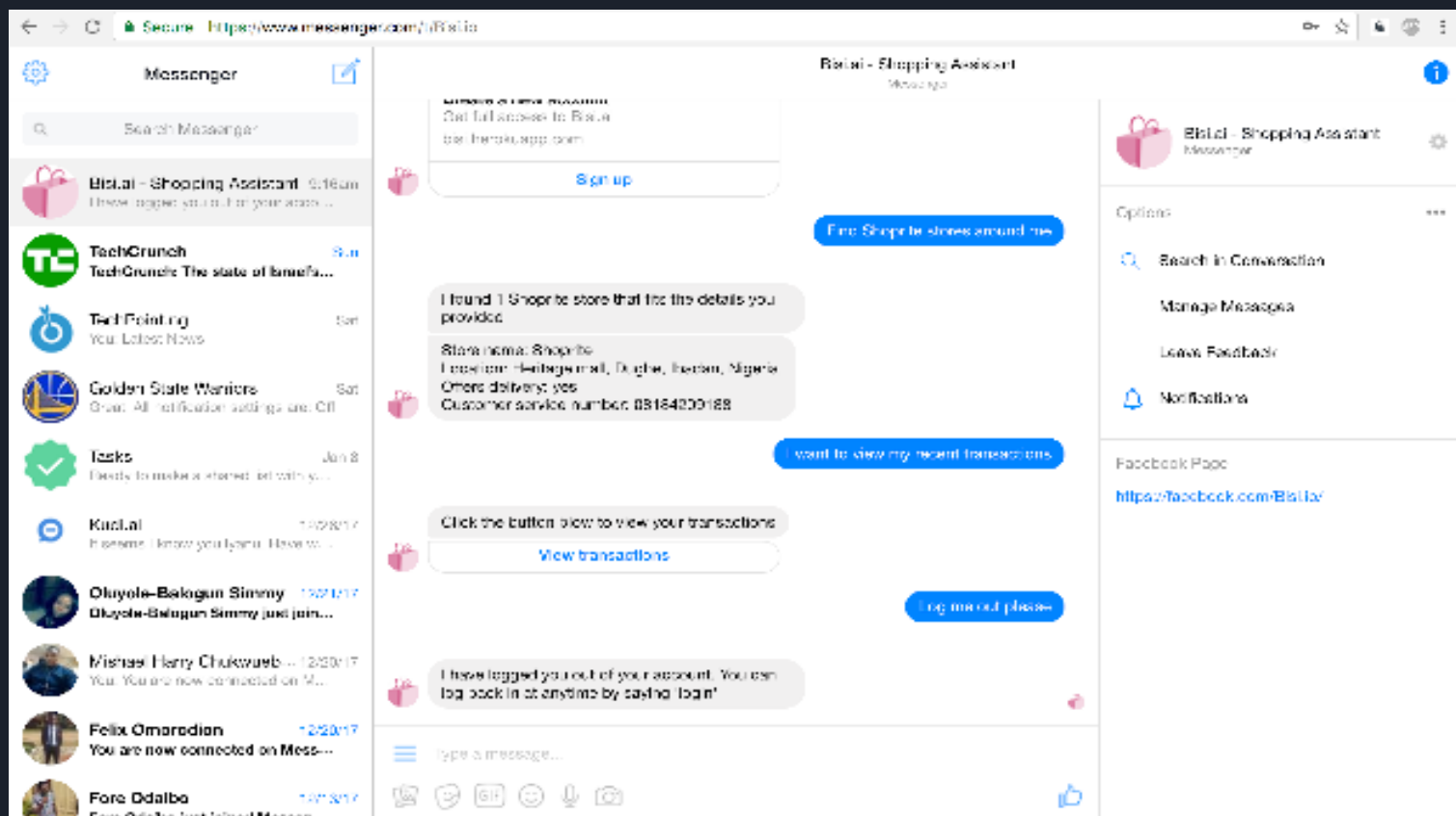
Account Logout

Figure 23:   A user logging out of his account

References

Sommerville, I. (2011). *Software Engineering, 9th Edition.*

Bala N. & Kumar S. (2016). E-Commerce And Customer Relationship Management. *IOSR Journal of Business and Management (IOSR-JBM) volume 12* .

Valacich et al. (2012). Essentials of Systems Analysis and Design.

Hudson S. (2004). The SCOR Model for Supply Chain Strategic Decisions. Retrieved from https://scm.ncsu.edu/scm-articles/article/the-scor-model-for-supply-chain-strategic-decisions

Liao Z. & Cheung M. (1999). Internet-based e-shopping and consumer attitudes: an empirical study. Elsevier Science B.V.

Mall R. (1999). Fundamentals of Software engineering. *Prentice-Hall of India* .

Huang Z. & Benyoucef M. (2013). From e-commerce to social commerce: A close look at design features. *Elsevier, volume 12, issue 4* .

Zhou L. & Zang P. (2013). Social commerce research: An integrated view. *Elsevier, volume 12, issue 2* .

Poyry E., Parvinen P. & Malmivaara T. (2013). Can we get from liking to buying? Behavioural differences in hedonic and utilitarian Facebook usage. *Elsevier, volume 12, issue 4* .

Thank You!