מסמך תיעוד –Handy

מערכת השאלת חפצים

מגישים:

יניב קרים 205731193

אביב בדיאן 313550550

אוקטובר 2019

**תוכן עניינים**

תיאור המערכת והפונקציונליות 3

טכנולוגיות 4

מבני נתונים 5

ארכיטקטורה 7

מדריך לשימוש באפליקציה 8

מנהל מערכת 8

משתמש באפליקציה 11

# **תיאור המערכת והפונקציונליות**

Handy הנה מערכת להשאלת חפצים מחנות וירטואלית אשר מציעה למשתמשים מגוון חפצים להשאלה. ישנם שני סוגים של משתמשים במערכת, לקוח ומנהל. לקוח במערכת אינו דורש רישום כמשתמש במערכת אלא רק צריך להוריד את האפליקציה ולהשתמש בה להנאתו. לעומתו, מנהל מערכת נדרש להזדהות בעזרת שם משתמש וסיסמה מוגדרים מראש על מנת לגשת לאפשרויות הניהול.

**לקוח:**

כאשר לקוח פותח את האפליקציה הוא רואה את קטלוג הפריטים אשר קיימים להשאלה בחנות, במידה והוא רואה פריט שברצונו לשאול, הוא בוחר בו ועובר למסך בו הוא בוחר מאיזה סניף פיזי לקחת את הפריט ותאריך שבו ברצונו לקחת אותו. לאחר מכן, על הלקוח למלא את הפרטים האישיים שלו (שם, מייל ומספר פלאפון) ולהמתין שמנהל יאשר / ידחה את בקשתו. הלקוח יקבל על כך הודעת SMS ייעודית מהמערכת.

**מנהל:**

לאחר שמנהל מתחבר עם פרטים מזהים ועובר למסך הניהול, באפשרותו לבצע מספר פעולות:

1. הוספת/מחיקת חפצים: מנהל יכול להוסיף חפץ חדש למערכת ולהעלות אותו עם תמונה שתשמר בשרת. כמו כן הוא יכול להסיר חפצים מהמערכת, פעולה אשר תגרור דחייה אוטומטית של כלל הבקשות הנוכחיות להשאלה של פריט זה.
2. הוספת/הסרת זמינויות של חפצים:לכל חפץ יש תאריכים בהם הוא זמין. זוהי רשימה של טווחי תאריכים בהם ניתן לשאול את החפץ. באפשרות המנהל להוסיף ולהסיר תאריכי זמינות עבור כלל הפריטים במערכת. תאריכים שכבר הושאל בהם הפריט או שהוזנו כבר כתאריכים אופציונליים להשאלה עבור פריט זה, יוצגו באדום (לא ניתן לסמנם שוב). הסרת תאריכים מסוימים תדחה באופן אוטומטי בקשות נוכחיות של השאלת הפריט בזמנים אלו.
3. אישור הזמנות של לקוחות: כאשר לקוח מבקש להשאיל חפץ, מנהל צריך לאשר לו את ההזמנה. במסך אישור הזמנות המנהל רואה את כלל הבקשות הפתוחות ובאפשרותו לאשר ולדחות אותן בהתאם לשיקוליו. על כל בחירה שעשה המנהל בנוגע לבקשה כלשהי (בין אם אישור ובין אם דחייה), הלקוח יקבל הודעת SMS שמעדכנת אותו באישור / דחיית בקשתו.

# **טכנולוגיות**

הטכנולוגיות שנבחרו לכל חלק במערכת הן:

1. **בסיס נתונים:** נבחר בסיס הנתונים הטבלאי PostgreSQL
2. **צד שרת:**
   1. גישה לבסיס הנתונים: דרייבר JDBC שמתחבר לPostgreSQL
   2. תקשורת שרת לקוח: שרת Spring אשר חושף ממשק REST כך שהלקוח יכול לפנות בHTTP לבקשות השונות
   3. קונפיגורציה: ההגדרות השונות של השרת נשמרות בקובץ XML אשר נטען. לדוגמה כתובת בסיס הנתונים ופרטי ההתחברות
3. **צד לקוח:** קליינט אנדרואיד
   1. שמירת תמונות: תשתית **Firebase-storage** של גוגל אשר מספקת שירותי אחסון.
   2. טעינה ותצוגת תמונות: ספריית **Glide** אשר מספקת יכולות טעינה ו-Caching של תמונות.
   3. גישה לשרת: ספריית **Retrofit** אשר מספקת שירותי Http client וממפה API Rest לממשק Java
   4. תשתית פיתוח אנדרואיד: **AndroidX** ספריות הרחבה עבור ספריות התשתית הישנות של אנדרואיד מעל Java.
   5. ניהול ממשק תאריכים: **savvisingh/DateRangePicker**- תצוגת Calendar בעלת תצוגה משופרת ויכולות מורחבות.
   6. הזרקת תלויות אובייקטים תצוגתיים: **Butterknife** – הזרקת אובייקטים ויזואליים לאובייקטים מאחורי הקלעים בצורה נוחה.
   7. ואלידציות: **saripaar** – מאפשרת ואלידציה אינטואיטיבית עבור שדות טפסים.
   8. העברת נתונים: **gson** – ספריית Serialize/Deserialize של אובייקטי Java ל-Json ולהיפך.

# **מבני נתונים**

מבני הנתונים במערכת כפי שהם נשמרים בבסיס הנתונים, בשרת ובלקוח (מפתחות ראשיים **בבולד וקו תחתון**, מפתחות זרים בבולד**)**:

1. Admins(**username,** password) – פרטי המנהלים השונים במערכת
2. Items(**id,** title, category, description, path) - החפצים השונים שניתנים להשאלה במערכת, יחד עם הקטגוריה, התיאור והנתיב לתמונה שלהם
3. Branches(**title**, address) – הסניפים השונים מהם ניתן לאסוף את החפצים והכתובת שלהם
4. Availabilities(**id**, **item\_id**, start\_date, end\_date) – זמינות עבור חפץ כלשהו בטווח תאריכים מסויימים
5. Borrows(**id**, **availability**, **branch**, phone, email, first\_name, last\_name, status)בקשות השאלה לזמינות מסוימת (של חפץ מסוים) בסניף מסוים יחד עם פרטי המבקש וסטטוס הבקשה, כאשר סטטוס יכול להיות
   1. Pending - ממתין לאישור או דחייה של מנהל
   2. Approved – בקשה מאושרת
   3. Declined – בקשה נדחתה

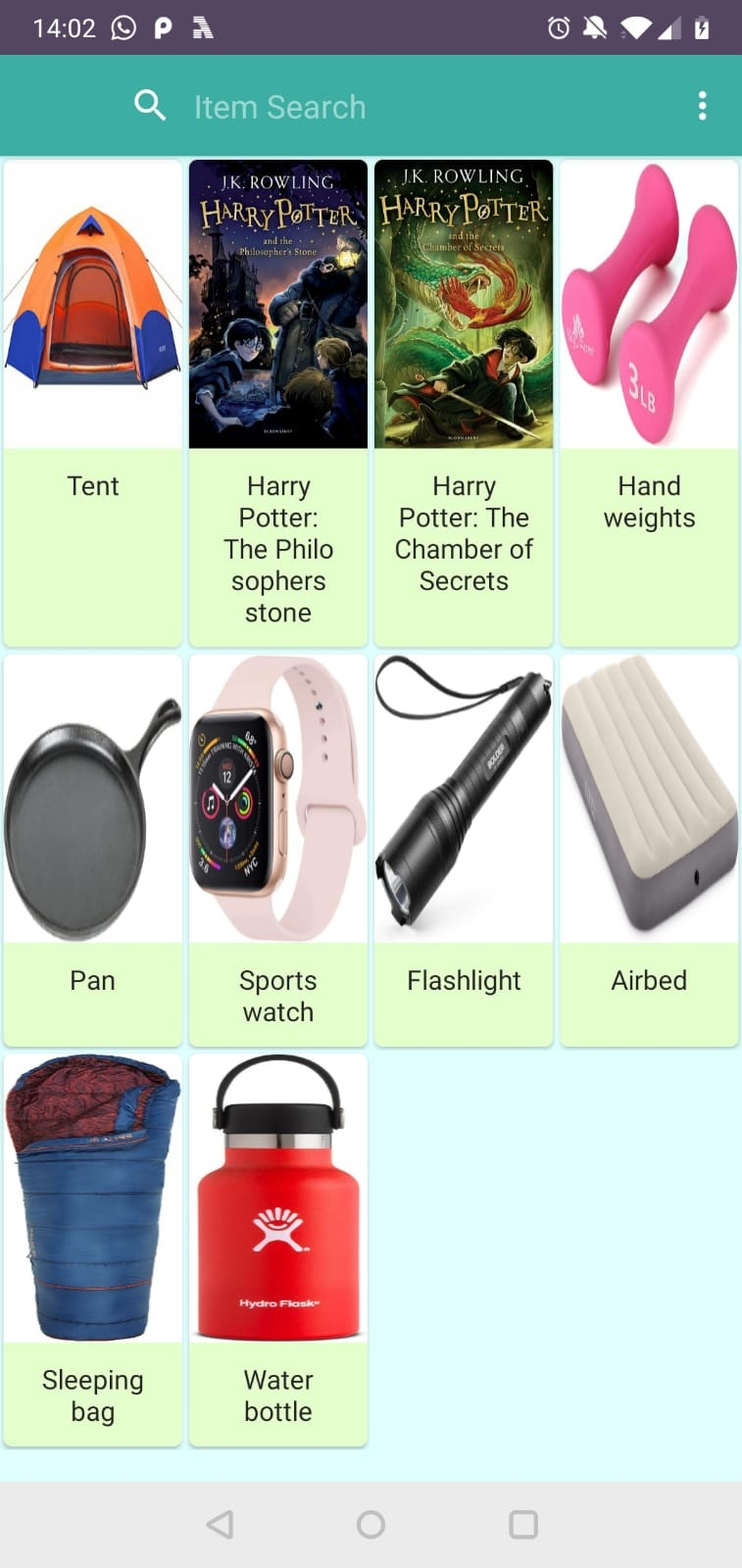
# **ארכיטקטורה**

1. Server side:
   1. Application: Main class that starts the spring application which start all the REST controllers
   2. AdminController: A controller which exposes REST api to log in to management
   3. AvailabilityController: A controller which exposes REST api to read, create and delete availabilities
   4. BorrowsController: A controller which exposes REST api to read, create and change the status of borrow requests
   5. BranchesController: A controller which exposes REST api to read, create and delete branches
   6. ItemsController: A controller which exposes REST api to read, create and delete items
   7. DataStore: A class which is responsible to create the connection to the database and make queries.
   8. DBConfig: A class that reads the system config XML in order to expose the connection data to the database
   9. ViewModels: A package that contains all the view models used in the systems
2. Client Side:
   1. MainActivity – starts the application. Responsible of displaying fragments that are related to the clients in the process of searching and requesting an item to borrow.
      1. ItemsDisplayFragment – Displays the items which a user can request to borrow. Items are loaded from the web server and can be queried through the search mechanism. After selecting item, the user can continue to the next fragment:
      2. BorrowReqeustFragment – Displays the selected item details and allows the user to pick dates in which he wants to borrow the item, as well as branch from which he wants to take the item. After picking dates and branch the user can continue to the next fragment
      3. BorrowConfirmFragment – Displays the complete request details, and responsible for asking the user personal information to complete the request.
   2. AdminLoginActivity – This activity is started after pressing the 'Admin' option in the main menu through a specified intent. It requests for admin login details, validates them, and after the server approves, the management activity is started.
   3. AdminManagementActivity – Responsible of the capabilities of admins. It contains a view page adapter – which shows each fragment in its own tab and enables scrolling between the different fragments.
      1. ItemsAdministrationFragment – Abilities for admin to add or remove items from the system.
      2. AvailabilitiesAdministrationFragment – Abilities for admin to add or remove availability for an item.
      3. RequestsAdminstrationFragment – List all the pending requests and the admin can decline or approve them.
   4. Models – A package with all the application models.
   5. RestAPI – Responsible for creating HTTP-client and send HTTP requests to the web server.

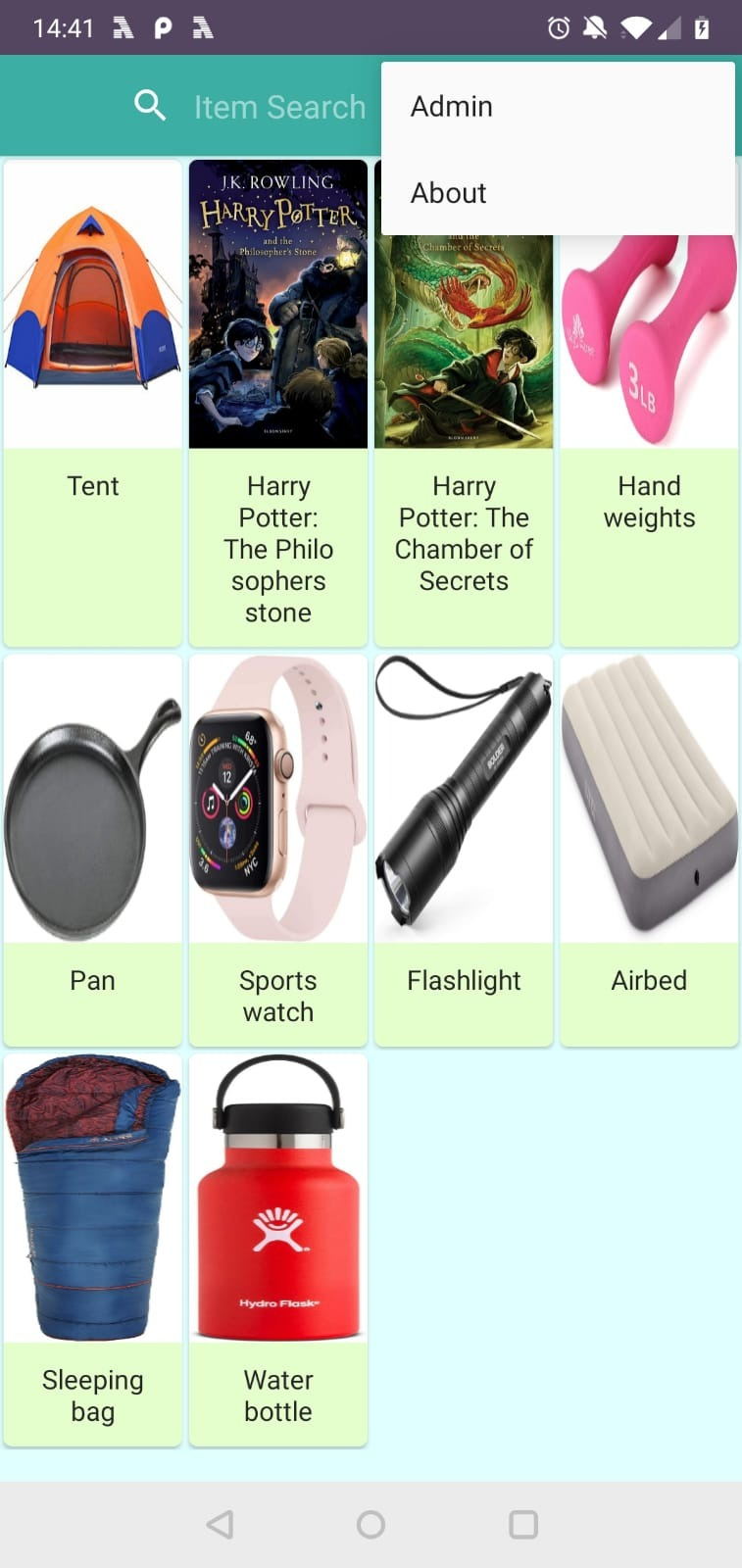
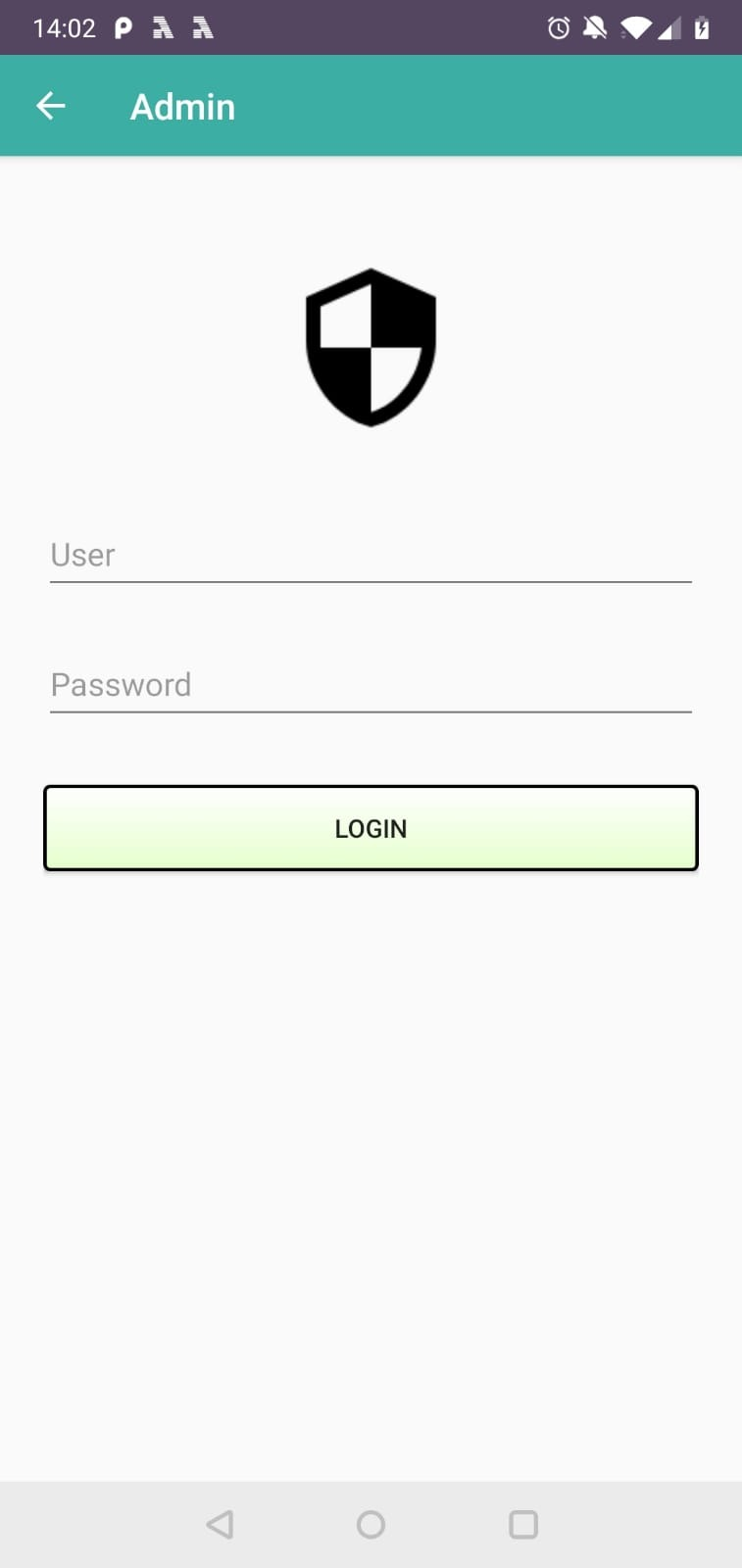
# **מדריך לשימוש באפליקציה**

## **מנהל מערכת**

ראשית, נפתח המסך הראשי של האפליקציה:



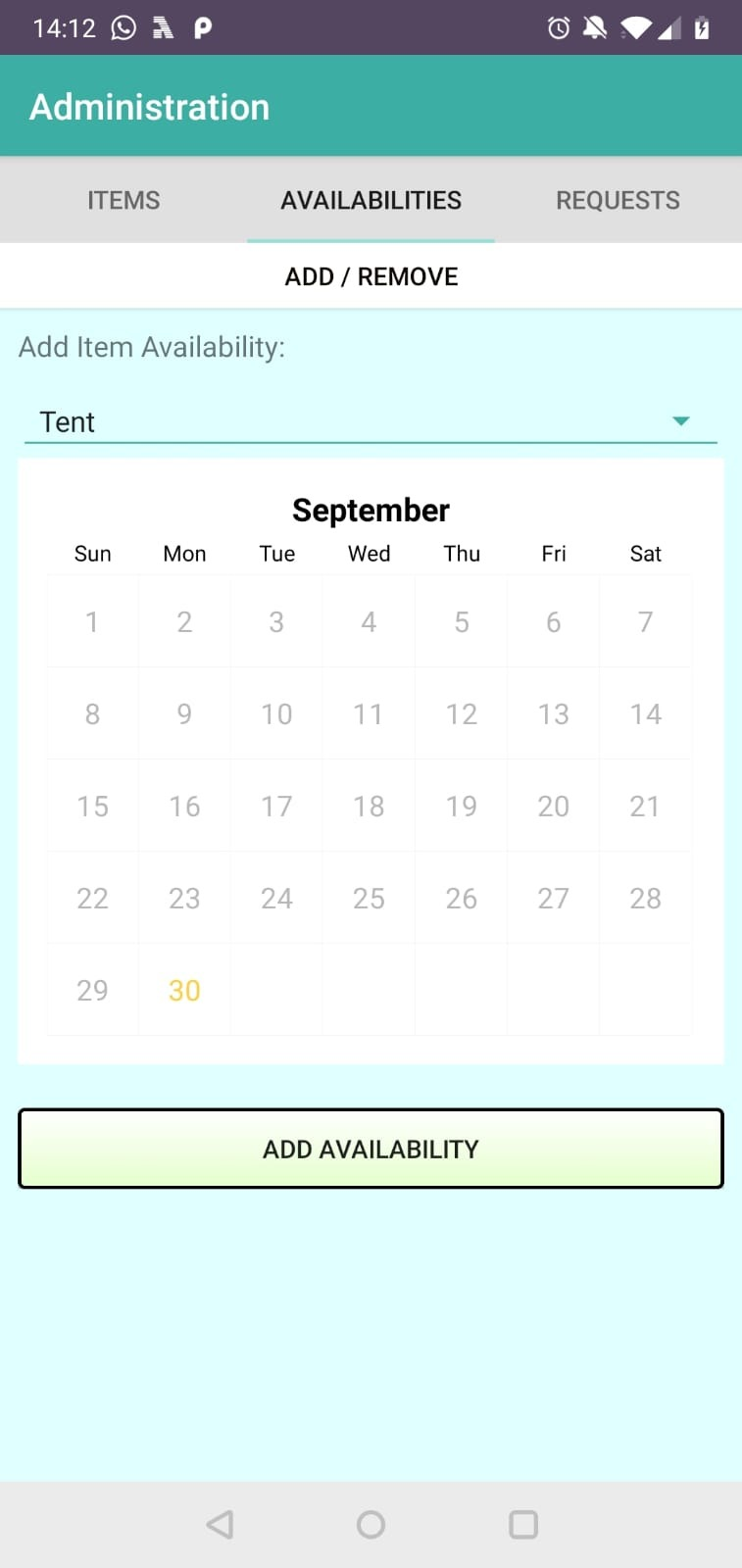
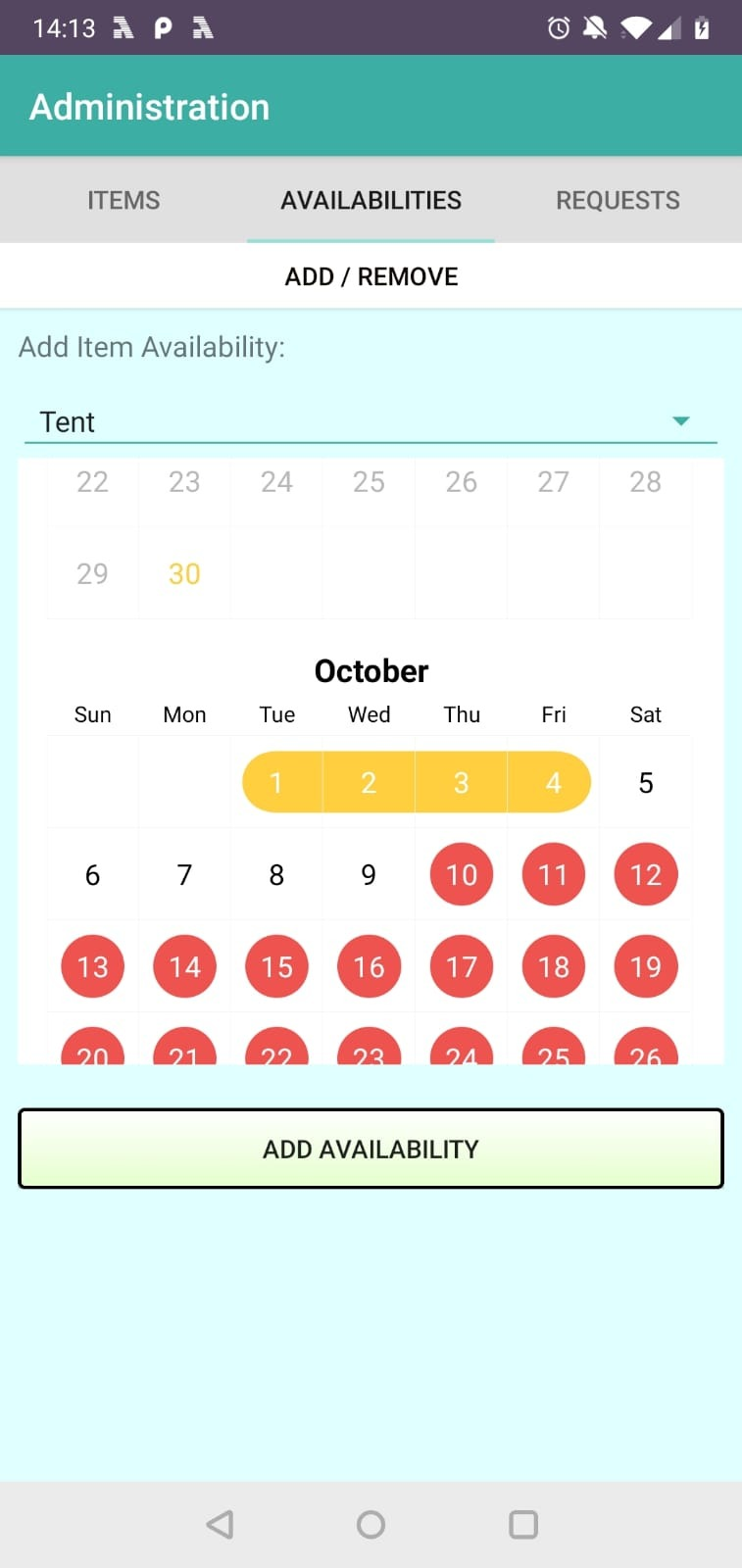
לחיצה על 3 הנקודות שיש בתפריט העליון מצד ימני תפתח תפריט.



לחיצה על admin בתפריט זה תוביל את המשתמש למסך התחברות.   
על מנהל המערכת להזין שם משתמש וסיסמה על מנת להכנס לממשק ניהול המערכת.

הערה: מנהלי המערכת ידועים ומוכרים מראש ושמורים במסד הנתונים.

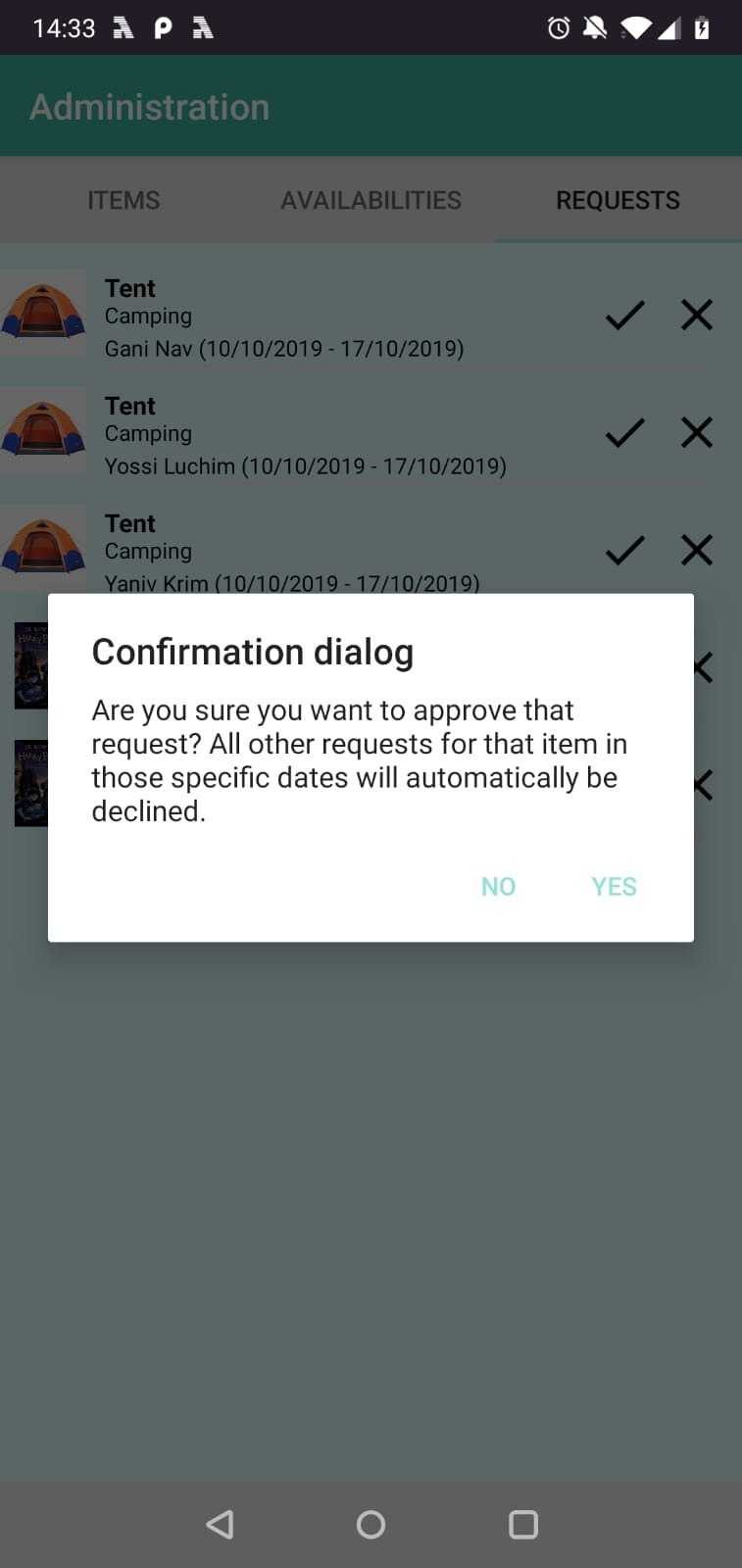
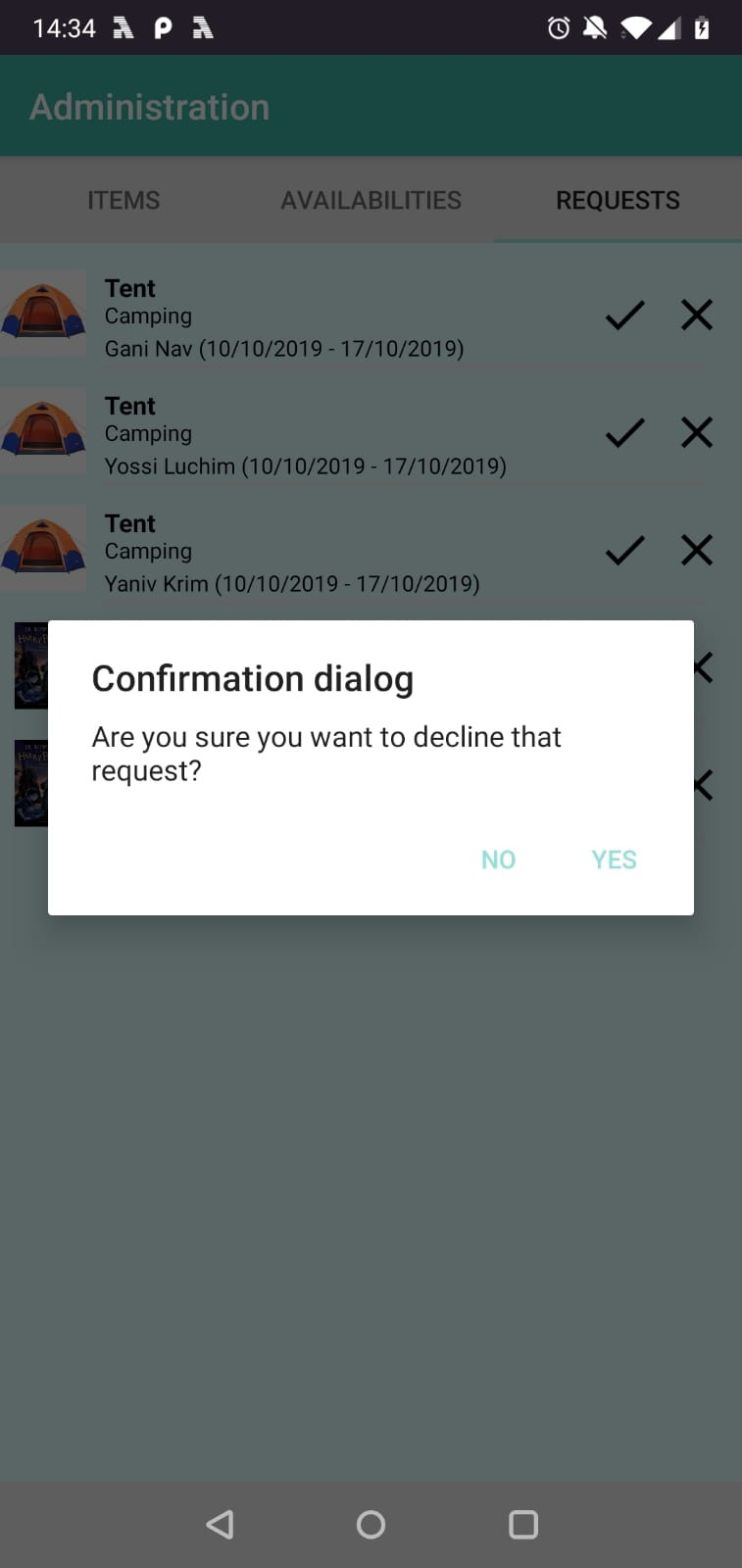
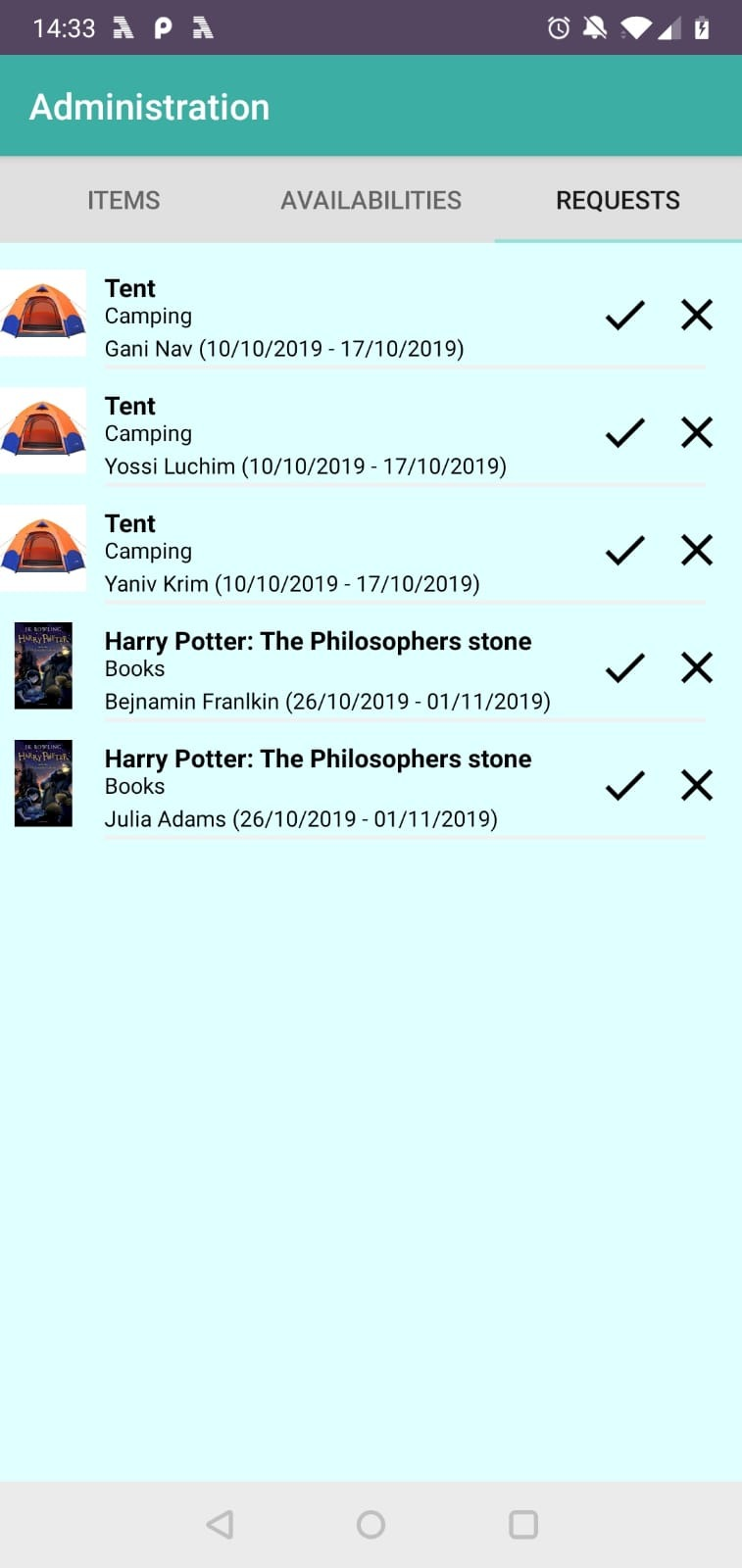
לאחר שהוזנו הפרטים המזהים של מנהל המערכת ובמידה ואלו היו נכונים, נפתח ממשק ניהולי של חפצים/זמינויות של פריטים /בקשות של לקוחות המעוניינים לשאול פריטים אשר מיוצג ע"י שלושה טאבים ואפשרות גלילה ביניהם (ITEMS, AVAILABILITIES, REQUESTS).

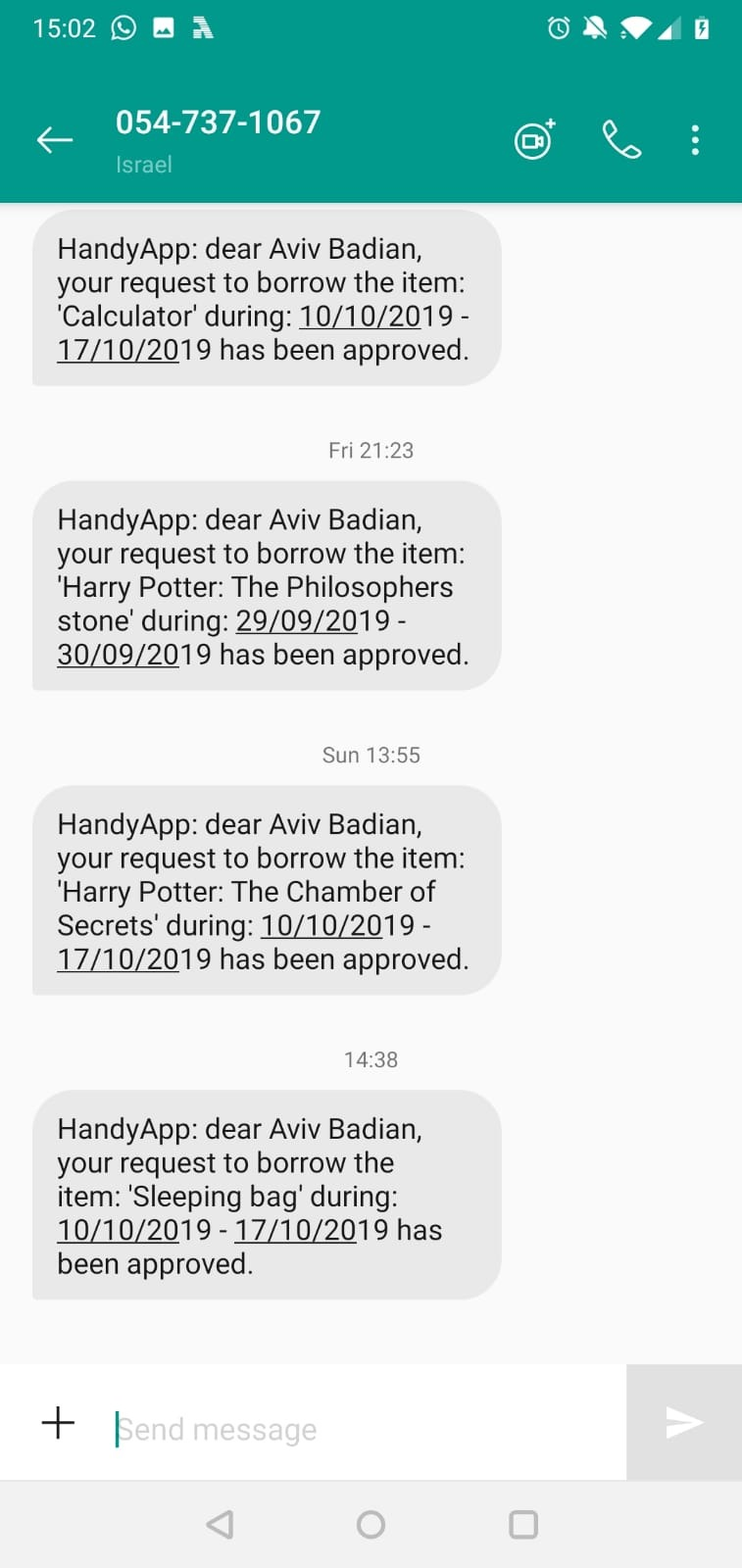
ראשית מוצג מסך הוספת הפריטים למערכת (המסך משמאל), בו יכול מנהל המערכת להזין פריט חדש. על מנת להזין פריט חדש יש למלא את כלל הפריטים המבוקשים- תמונה, שם פריט, קטגוריה אליה משוייך הפריט ותיאור.

גלילת המסך ימינה תעביר את המנהל לsection של הוספת זמינויות לפריטים (המסך האמצעי)- באפשרות המנהל להוסיף תאריכים בהם הפריט פנוי להשאלה. תאריכים אדומים מסמנים תאריכים בהם המנהל כבר אישר השאלה של פריט זה לאדם אחר, או שכבר הזין תאריכים אלו כתאריכים בהם הפריט יוכל להיות מושאל בעתיד, ולכן לא יכול לבחור בהם כעת כזמני זמינות של הפריט.   
התאריכים אותם בוחר המנהל יסומנו בצהוב החל מתאריך ההתחלה עליו הוא לוחץ ראשון ועד לתאריך הסיום (מסך ימני).

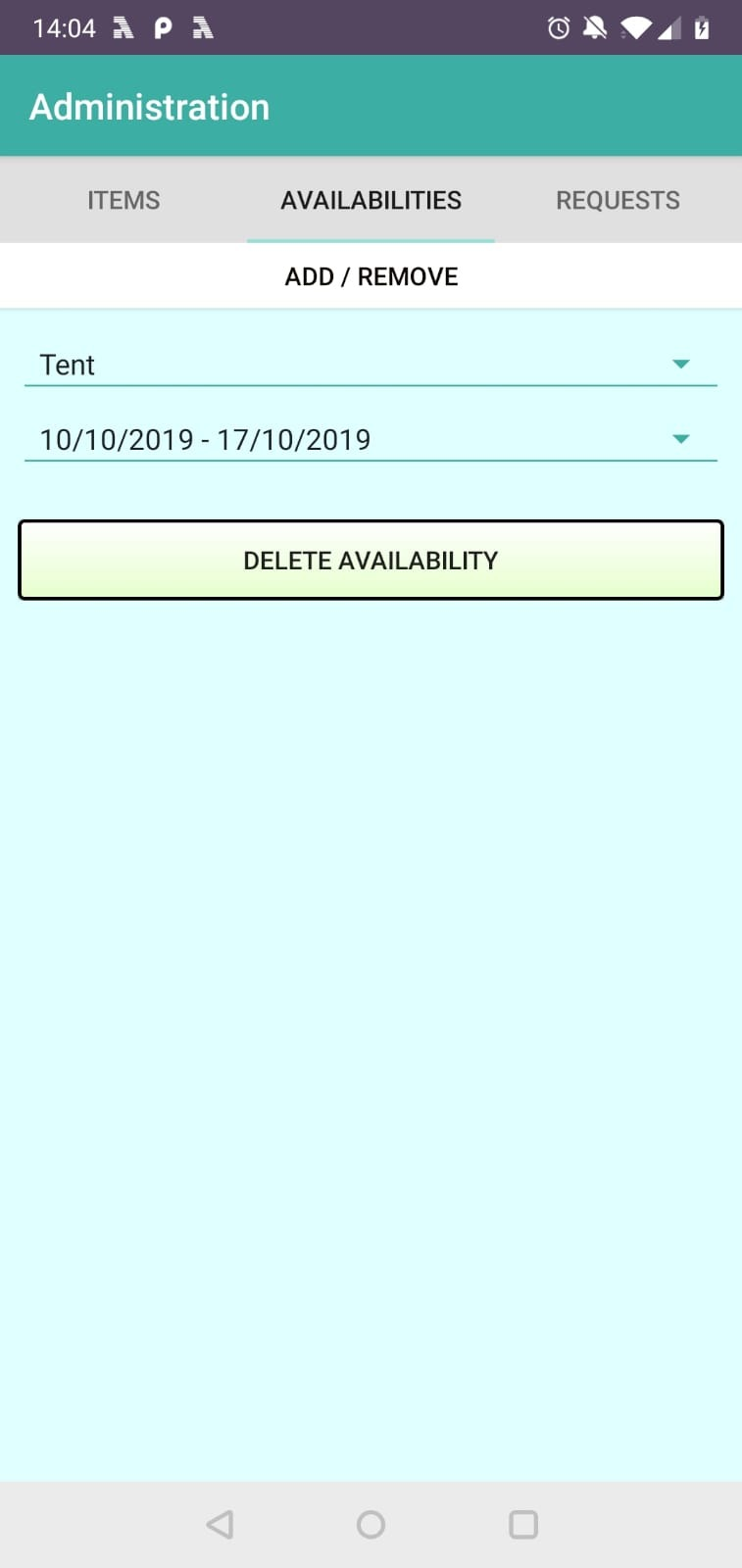
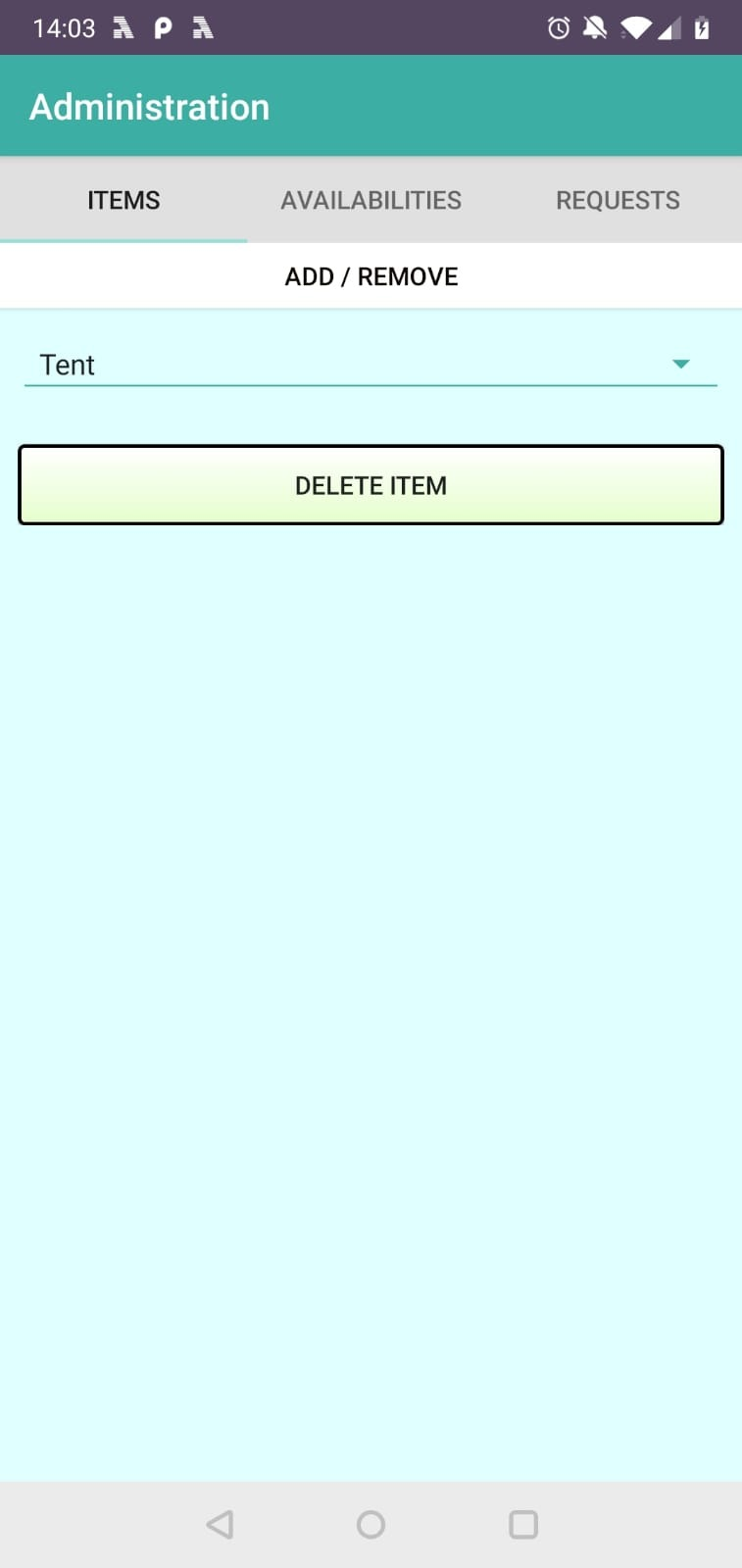
גלילת המסך ימינה פעם נוספת תעביר את המנהל ל-section של אישור או דחיית בקשות השאלה:

במסך זה יכול המנהל לראות את הפריט אותו ביקשו לשאול, מי מבקש את הפריט ובאילו תאריכים. לחיצה על סימן האיקס לצורך דחיית הבקשה תפתח את המסמך האמצעי המוודא עם המנהל שאכן הוא רוצה לדחות את הבקשה. לחילופין, לחיצה על סימן הוי לצורך אישור ההשאלה תפתח את המסך הימני, המוודא שהוא אכן מאשר הזמנה זו ובכך דוחה את ההזמנות האחרות לפריט בתאריכים אלו. במידה והמנהל השיב yes, נשלחת ללקוח הרלוונטי הודעת SMS שמיידעת אותו על פרטי ההשאלה:

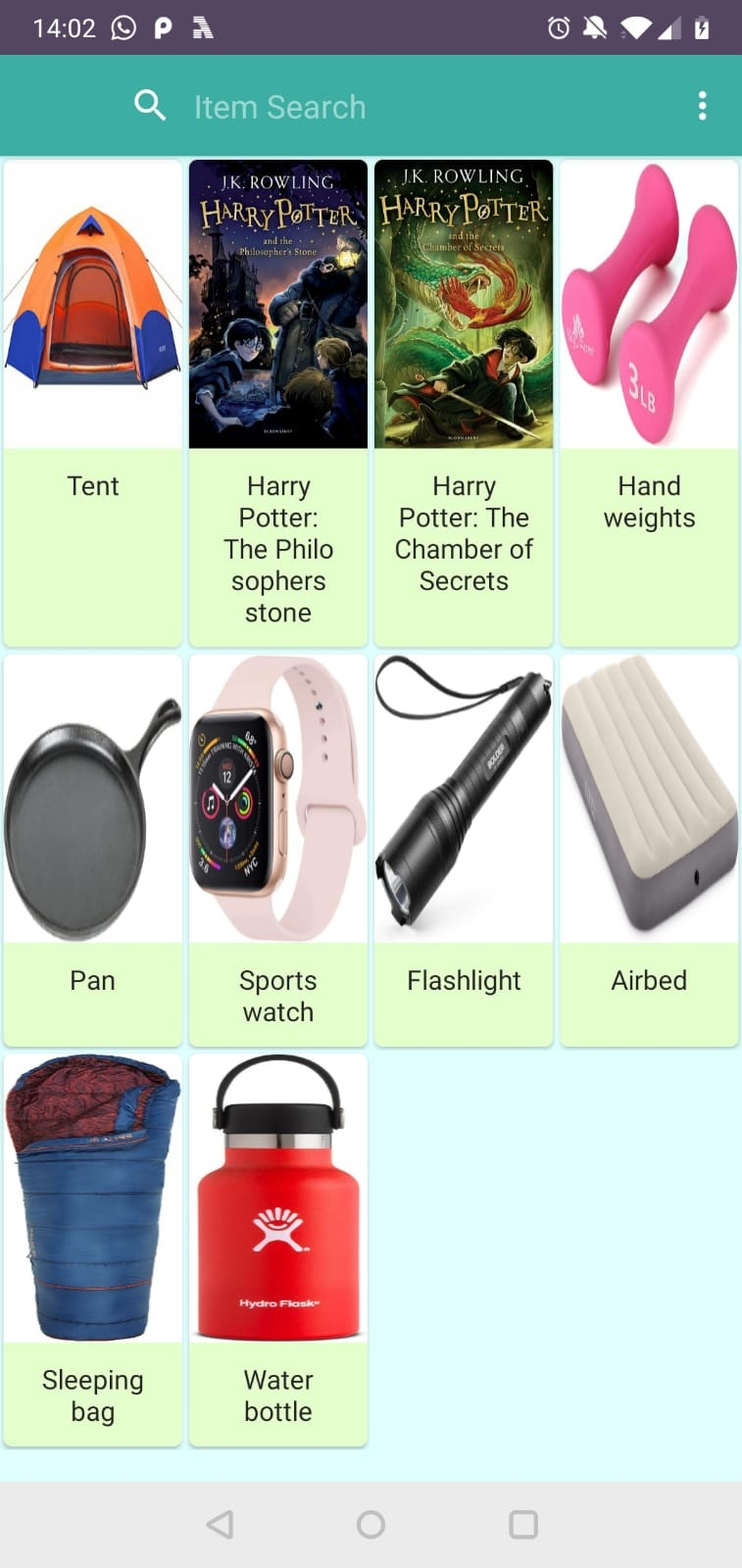


בשני ה-sections: ITEMS, AVAILABILITIES ישנה אפשרות ע"י לחיצה על כפתור ה- ADD/REMOVE שבחלק העליון של המסך לעבור בין אפשרות של הוספה ומחיקה של פריט או זמינות, בהתאמה:

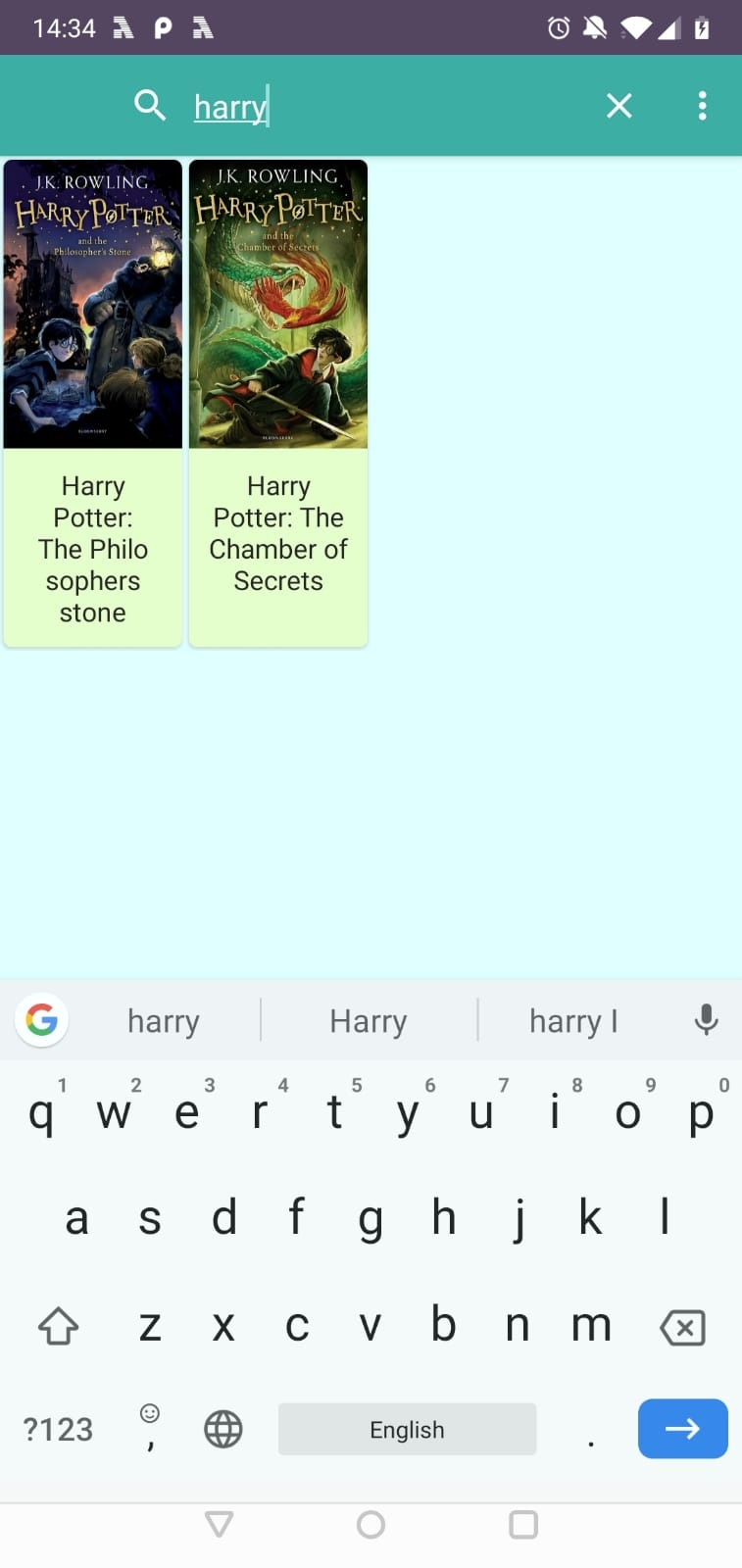
## **משתמש באפליקציה**

ראשית נפתח המסך הראשי:

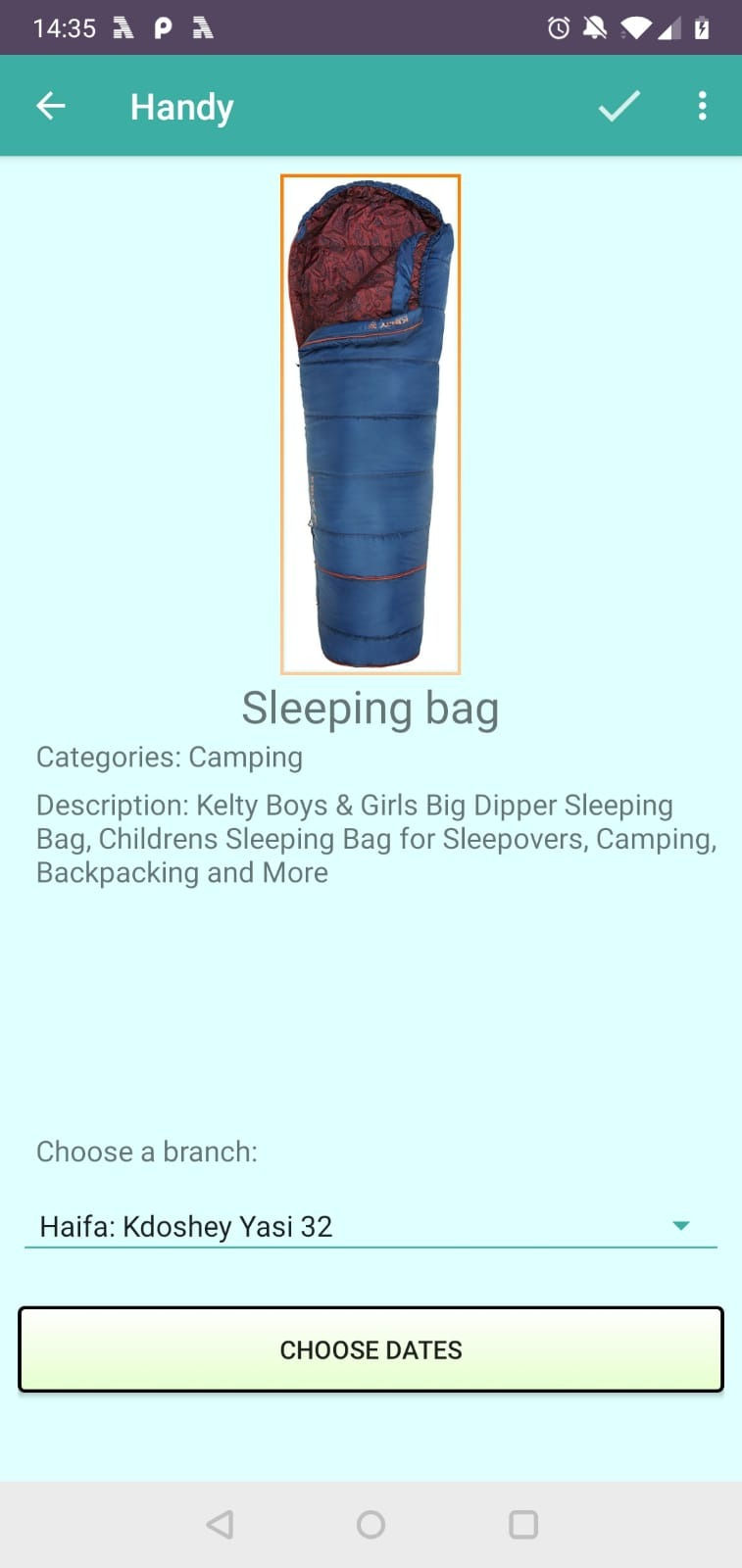


בפני המשתמש מוצגים כלל הפריטים שקיימים במערכת להשאלה. כפי שניתן לראות כל פריט מוצג בעזרת תמונה ושם פריט.

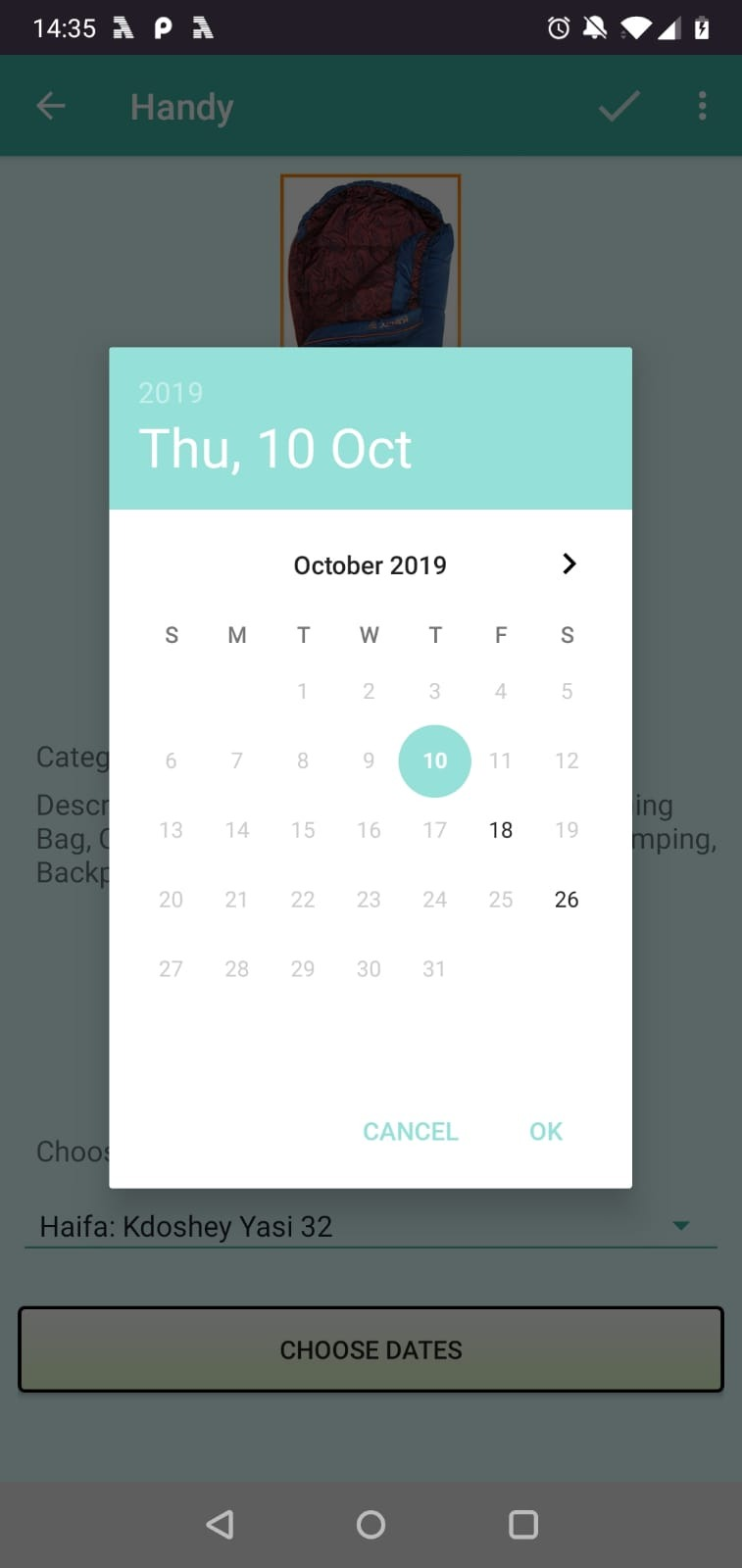
באפשרות המשתמש לבצע חיפוש בין שלל הפריטים ע"י חיפוש שם של פריט או שם חלקי באמצעות הזנת טקסט בחלק העליון של המסך בסמוך לזכוכית המגדלת:



לחיצה על פריט כלשהו פותחת למשתמש עמוד ייעודי של הפריט ובו מוצגים כלל הנתונים אודותיו.   
כפי שניתן לראות בתמונה שלהלן, המשתמש יכול לבחור סניף ממנו הוא מעוניין לאסוף את הפריט.

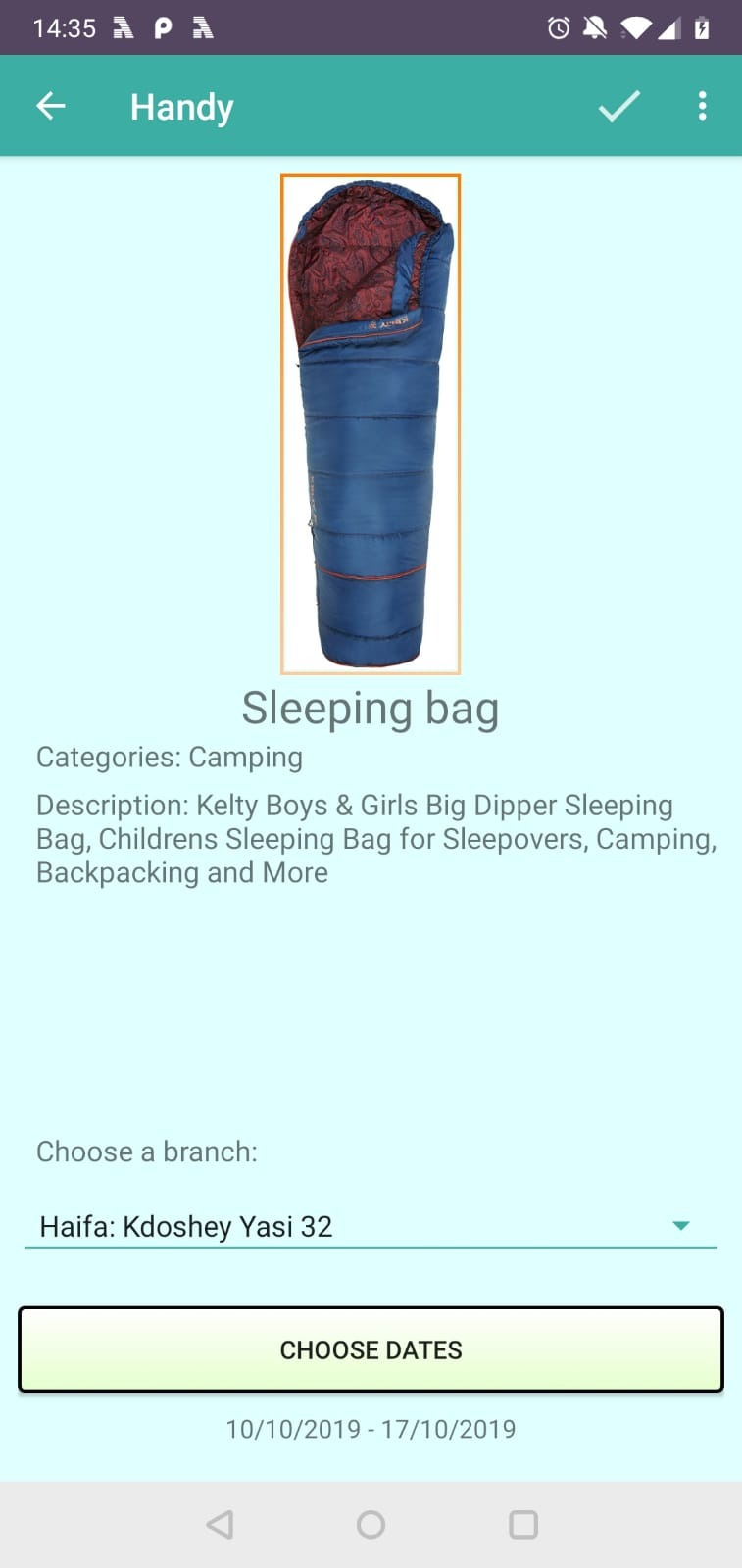


לחיצה על CHOOSE DATES תציג בפני המשתמש לוח שנה עם התאריכים הפנויים להשאלת הפריט:

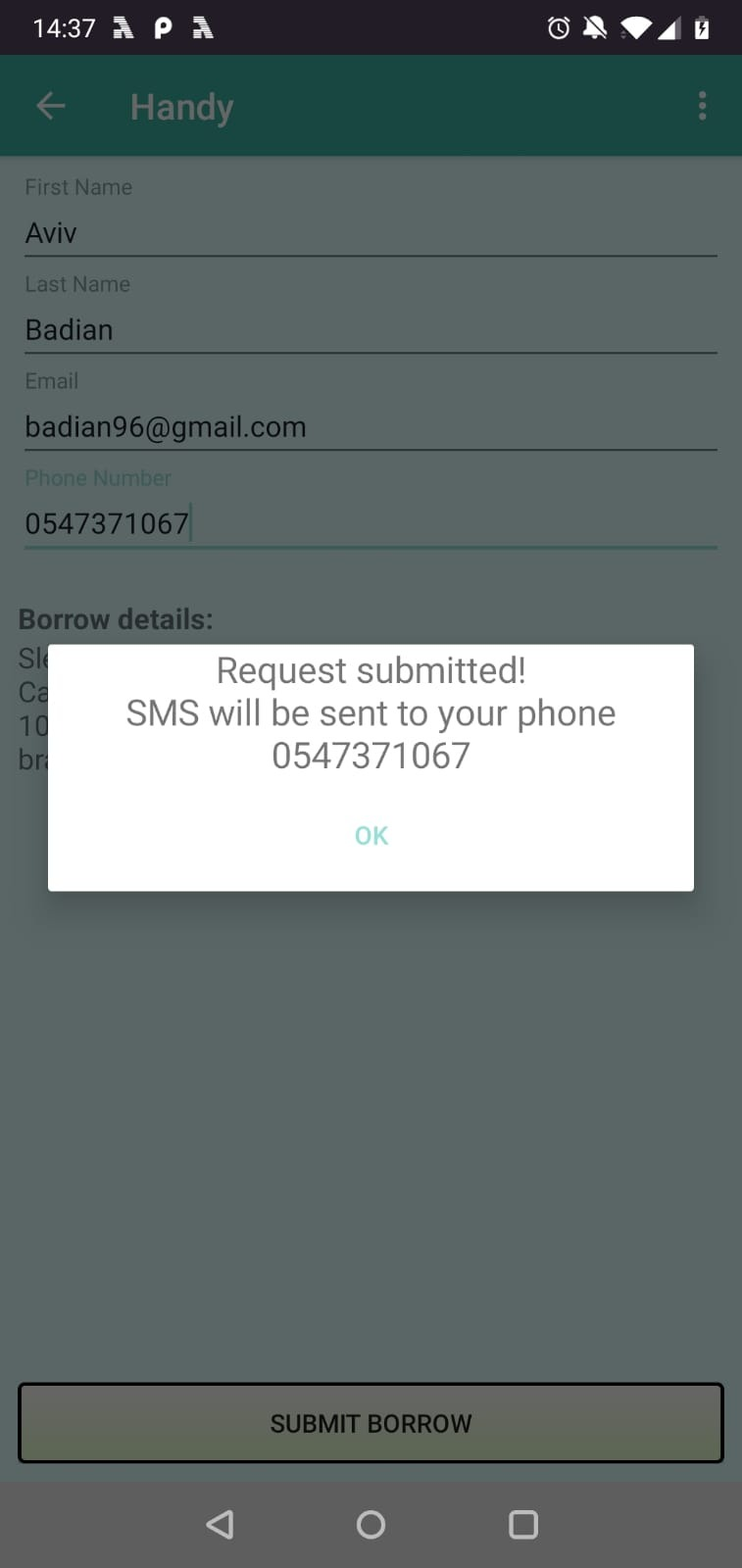
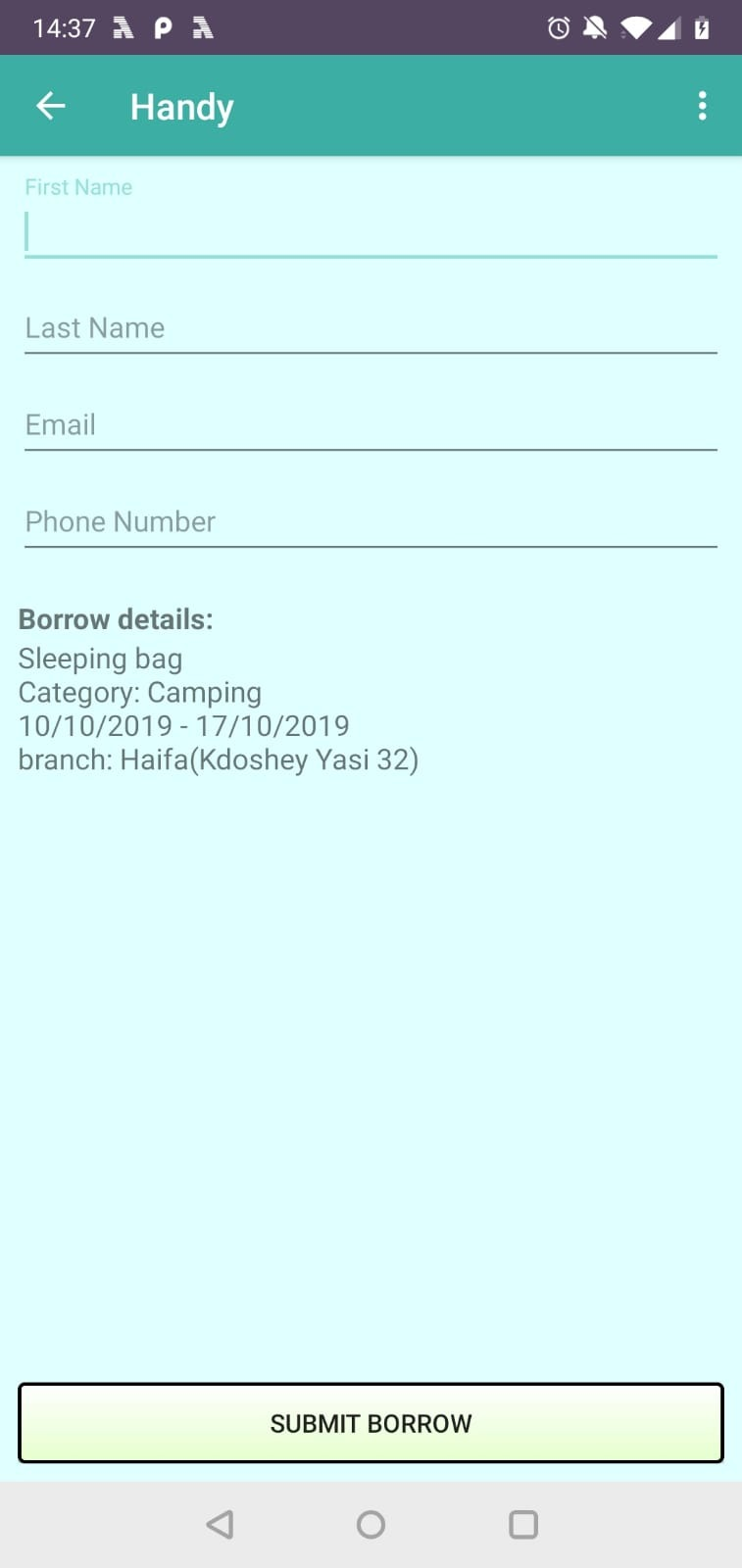


ניתן לראות במקרה זה כי רק בתאריכים המובלטים (10,18,26) בחודש זה הפריט זמין וניתן להשאלה. שאר התאריכים מאופרים ולכן אינם לחיצים לצורך בחירתם. על המשתמש לבחור תאריך לצורך תחילת השאלה.

לאחר בחירת התאריכים ואישורם, יחזור המשתמש למסך הפריט, כשבתחתית המסך יוצג תאריך תחילת ההשאלה כפי שבחר, ובנוסף יוצג תאריך היעד בו הוא נדרש להחזיר את הפריט.



כל שנותר כעת למשתמש לעשות הוא ללחוץ על וי במעלה המסך לצורך אישור ההשאלה. המשתמש יועבר למסך שבו עליו להזין את פרטיו האישיים:

המשתמש חייב להזין את כלל הפרטים הנדרשים, ולאחר מכן לחיצה על SUBMIT תשלח את בקשתו להשאלה אל מנהלי המערכת לאישור / דחייה ותציג לו הודעת אישור.